A Redundant Parallel Robotic Machining Tool: Design, Control and Real-Time Experiments

Hussein Saied, Ahmed Chemori, Micaël Michelin, Maher El Rafei, Clovis Francis, François Pierrot

A Redundant Parallel Robotic Machining Tool: Design, Control and Real-Time Experiments

H. Saied, A. Chemori, M. Michelin, M. El Rafei, C. Francis and F. Pierrot

Abstract In this chapter, we present a machining device, named ARROW, designed with the architecture of a redundant parallel manipulator capable of executing five degrees-of-freedom in a large workspace. Machine-tools based on parallel robot development are considered a key technology of machining industries due to their favourable features such as high rigidity, good precision, high payload-to-weight ratio and high swiftness. The mechanism of ARROW robot isolates its workspace from any type of inside singularities allowing it to be more flexible and dynamic. An improved PID with computed feedforward controller is implemented on ARROW robot to perform real-time experiments of a machining task. The control system deals with antagonistic internal forces caused by redundancy through a regularization method, and achieves a stability conservation in case of actuators saturation. The results are evaluated using a root mean square criteria over all the tracking error confirming the high accuracy and good performance of ARROW robot when used for machining operations.

1 Introduction

Machining is the process of a controlled material-removal that makes a desired deformation in the shape and size of raw materials. The theme of material-removal

Hussein Saied, Ahmed Chemori and Francois Pierrot
LIRMM - UM, 161 rue Ada, 34095 Montpellier cedex 5, France, e-mail: {saied,chemori,pierrot}@lirmm.fr

Micael Michelin
TECNALIA France, Centre Spatial Universitaire, Btiment 6, 950 Rue de Saint Priest, 34090 Montpellier - France, e-mail: micael.michelin@tecnalia.com

Hussein Saied, Maher El Rafei and Clovis Francis
CRSI - UL, Rafic El Hariri Campus, Faculty of Engineering, Beirut, Lebanon, e-mail: {maher.elrafei,cfrancis}@ul.edu.lb

classifies machining in the subtractive manufacturing classes [6]. It is considered as an essential part in the manufacturing of many products based on metal, wood, plastic, ceramic, and composites. The conventional term, ”machining”, refers to a set of processes such as turning, boring, drilling, milling, broaching, sawing, shaping, planing, reaming, tapping ... etc [3]. Nowadays, most of the recently mentioned operations are carried out by Computer Numercial Control (CNC) in which the computers are used to control the movement and act of shaping.

Three principal operations of machining are classified as turning, drilling and milling. All other operations are considered as miscellaneous categories described in what follows:

- Turning operation is performed by rotating the workpiece and facing it over the cutting tool as shown in Fig. 1(a).
- Milling operation is the operation in which the cutting tool rotates bringing the cutting edges to bear against the workpiece, see Fig. 1(b).
- Drilling operation is the most common machining process. It is the generation of cylindrical holes by bringing the rotating cutter with cutting edges into the workpiece as illustrated in Fig. 1(c).
- Miscellaneous operations are secondary swarf producing operations performed on the workpiece after the principal machining processes using the typical machining tools. For example: boring, broaching, burnishing ... etc.

The successful design of machine-tools is based on several requirements such as: high accuracy resulting in a best surface integrity, acceptable level of vibration, high speeds and feeds increasing the productivity, high static stiffness of the different machine-tool elements such as structure, joints, and spindles, avoidance of unacceptable natural frequencies that may cause resonance of the machine-tool, and finally adopting modern control techniques [27].

Traditional machine-tools are based on serial manipulators designed as a series of links connected via actuated joints from the base to the end-effector (see Fig. 2).

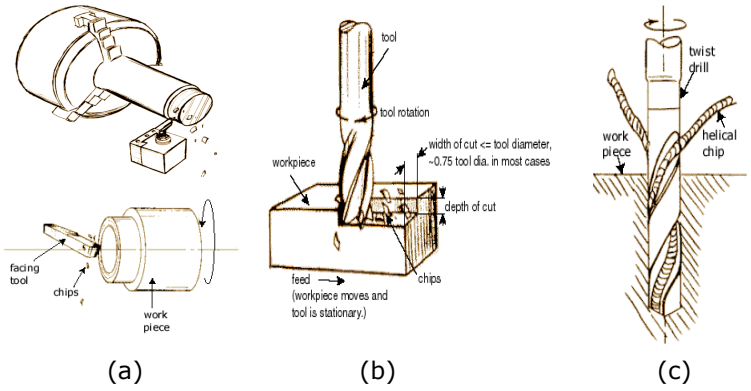


Fig. 1 Principal machining operations. **(a)** turning. **(b)** milling. **(c)** drilling [4].

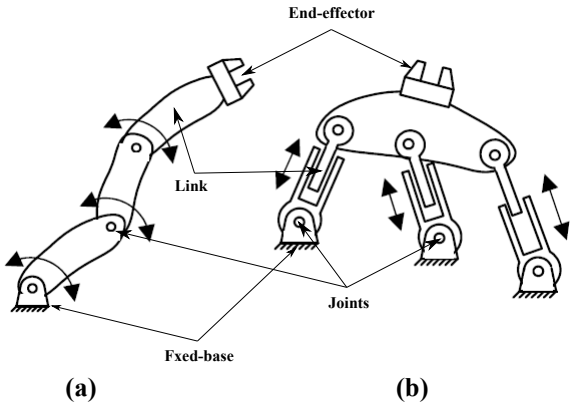


Fig. 2 Illustration of Kinematic manipulators. **(a)** serial manipulators. **(b)** parallel manipulators. [8].

These manipulators, known with their large range of motions, lack to the good accuracy and high stiffness. Moreover, they have low dynamic response and valuable level of vibration [8, 11].

Parallel Kinematic Manipulators (PKMs) offering many advantages over their serial counterparts are welcomed candidates to be the future machining robots. According to [11], any structure made up of a fixed base and traveling plate linked together with at least two independent kinematic chains is called a PKM (see Fig. 2). PKMs provide large accuracy, low vibration, high acceleration capabilities thanks to the light moving parts, and high stiffness due to the closed-chains structure. However, these manipulators are characterized by a limited workspace especially in the rotational motion. In the addition to their complexity in the forward kinematic solutions, and the considerable number of singularities inside and on the borders of their workspace [8, 11].

Due to the satisfying features of PKMs, they have achieved a real success in various applications other than machining. PKMs are considered the leading robots in the automated industrial applications, particularly in the pick-and-place food packaging tasks which require rapid execution maintaining an acceptable precision [18]. Also they are used in the field of laser cutting operations [1], medical applications (Robot-Assisted Surgery (RAS) [23], CardioPulmonary Resuscitation (CPR) [9]), flight simulators (Stewart platforms [24]), Haptic devices (delta robot [5]).

Beside their closed-chains structure which increases the stiffness, decreases the level of vibration and enhances the precision, PKMs shall be equipped with advanced control techniques to have the best performance of automation. Control of PKMs is considered as a very challenging task in the literature due to a set of factors including:

- PKMs are known by their higly nonlinear dynamics which may increase considerably when operating at high speeds.

- Actuation redudancy which means that the number of actuators is greator than the number of degrees of freedom [11] exists in some parallel robots.
- Uncertainties and variations of the dynamics and operating conditions are often present in PKMs.

Several control techniques have been proposed and studied in the literature starting from the non-model-based non-adaptive and adaptive controllers, ending up with model-based non-adaptive and adaptive controllers. One can mention few examples from the aforementioned classes of control implemented on PKMs respectively: Proportional-Integral-Derivative (PID) [29, 18], nonlinear PID [25], Augmented PD (APD) [28], Adaptive Feedforward plus PD [2, 17], Desired Compensation Adaptive Control Law (DCAL) [12], Adaptive controller based on the Robust Integral of the Sign of the Error (RISE) [13], Feedforward Compensation in L1 Adaptive Control [14].

This chapter introduces a five-degree-of-freedom redundantly actuated PKM named ARROW[1] characterized by the good compromise between rapidity and precision developed for machining tasks. In section 2, a general overview on ARROW PKM's structure is stated, in addition to the kinematic and dynamic modeling with a brief presentation of the analyzed singularities. Section 3 describes the machining task to be tested, the required trajectory generation, and the proposed control solution. Experimental results are illustrated, discussed and interpreted in section 4. Finally, section 5 concludes the chapter and provides future work suggestions.

2 ARROW PKM: General Overview, Modeling and Singularity Analysis

The ARROW robot is a redundantly actuated PKM has been designed and developed at LIRMM within the ARROW project[1]. This section covers the general description of ARROW PKM, the detailed inverse kinematic model, the differential kinematics and dynamics associated to ARROW robot, and a brief analysis for the singularities related to the operational workspace. Note that the reader can refer to [21, 22] for further information.

[1] ARROW project: a national french research project financed by the National Research Agency (ANR). Its main objectives can be summarized in the design of **A**ccurate and **R**apid **R**obots with large **O**perational **W**orkspace, from which the acronym **ARROW** has been derived. The project embraces three partners: IRCCyN (Institut de Recherche en Communication et Cyberntique de Nantes), LIRMM (Laboratory of Informatics, Robotics and Microelectroncs of Montpellier) and Tecnalia France.

2.1 General Overview of ARROW PKM

The structure of ARROW PKM consists of two separate modules as illustrated in Fig. 3: the parallel module and the turntable. The parallel module involves six linear actuators each pair lays in the same slider track and equidistant each other. This configuration facilitates the geometric calibration and reduces the overall cost from manufacturing point of view. These six actuators are linked by means of spherical joints to the kinematic chains which are attached to the articulated traveling plate by revolute joints (see Fig. 4). The idea behind using revolute joints is to avoid some possible internal collisions between the arms. The two intermediate parallelogram arms are responsible of the rotation of the traveling plate around the vertical axis parallel to the z-axis, while its position and orientation are originated from the settle of all the actuators together. Note that the traveling plate can reach $\pm 45^{\circ}$ as limits for its allowable range of rotation around z-axis. Thus, the prismatic motion of the linear actuators sets up the traveling plate in 3T-1R[2] dofs.

The turntable is placed facing the parallel module (of Fig. 3) and kept outside the severe zone of collision with the traveling plate. It is actuated with two rotating motors allowing it to turn around the axis parallel to the x-axis of the fixed frame shown in Fig. 4(a). It handles the object to be machined and provides the 5th rotational dof to the motion range of the traveling plate. It is equipped with a spring assembly, as illustrated in Fig. 5, contributing against gravitational effect and allows maintaining the turntable orientation fixed without any need to the motors torques. This is beneficial regarding energy saving and reduces control efforts.

From a control point of view, the main part to be controlled is the parallel module. It holds the machining tool represented as a traveling plate which is a milling machine in our case. Implementing robust control algorithms to the linear actuators will set up the machine-tool in a precise tracked trajectory for machining purpose.

2.2 Modeling of ARROW PKM

This part explains in details the kinematic, differential kinematic and dynamic models of ARROW robot. In view of the reported mutations in [22] that have been suggested to implement ARROW PKM avoiding internal arm collisions, the Forward Kinematic Model (FKM) is difficult to establish if not impossible. Thus, only the Inverse Kinematic Model (IKM) will be presented and clarified.

Fig. 6 shows the parallel module with some notations to be used in modeling. Six arms numbered from I to VI are grouped as simple and parallel arms. The lengths $L_i(i = 1,2,5,6)$ of the simple arms are equal to L_s, while lengths $L_i(i = 3,4)$ of the parallel arms are equal to L_p.

[2] "T" corresponds to translational motion and "R" corresponds to rotational motion.

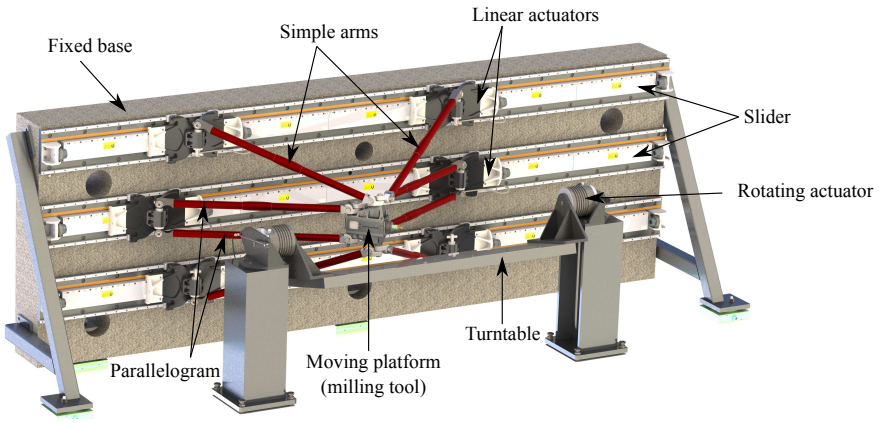


Fig. 3 CAD view of the ARROW machining PKM.

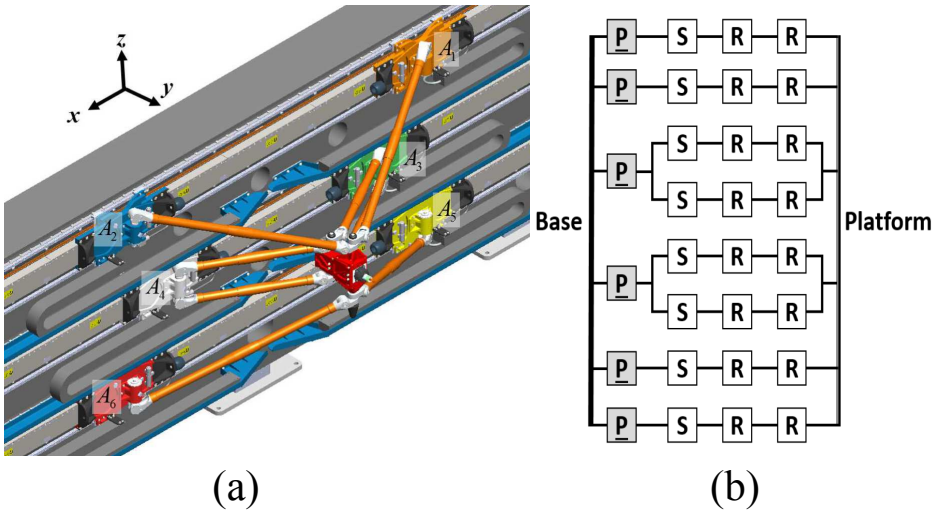


Fig. 4 ARROW PKM. **(a)** 3D CAD view of the parallel module. **(b)** graph diagram [22].

Two frames are defined for the modeling: Base frame (O, e_x, e_y, e_z) and Platform frame $(P, e_{xp}, e_{yp}, e_{zp})$ (see Fig. 9, Fig. 8). The following recorded points are coordinated in the base frame:

- $\boldsymbol{A_i} = (x_i \quad y_i \quad z_i)^T = (q_i \quad y_i \quad z_i)^T$ is the point attached to the spherical joint for $i = 1,2,5,6$ and the center of the linear actuator for $i = 3,4$. q_i is the position of the linear actuator in the x-direction $\forall i = 1,..,6$ (see Fig. 6).
- $\boldsymbol{B_i} = (x_{bi} \quad y_{bi} \quad z_{bi})^T \ \forall i = 1,..,6$ and $\boldsymbol{D_i} = (x_{di} \quad y_{di} \quad z_{di})^T$ for $i = 1,2,5,6$ are the points attached to the revolute joints at the level of the platform, and that is shown clearly in the schematic view of Fig. 9. For $i = 3,4$, $\boldsymbol{D_i}$ is centered between the two revolute joints of the platform on its y-axis (see Fig. 8).

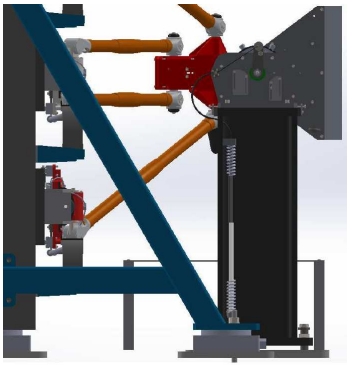


Fig. 5 Close side view to the CAD drawing of ARROW PKM showing the springs used in the turntable. [22].

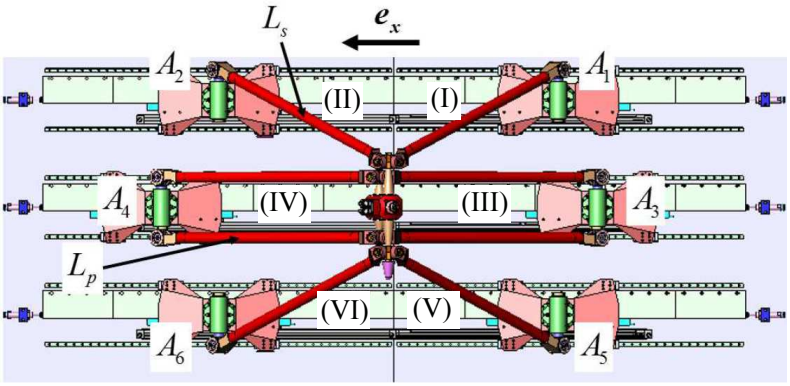


Fig. 6 Facing CAD view of the parallel module of ARROW PKM with some notations. [22].

- $\boldsymbol{C_i} = (x_{ci} \quad y_{ci} \quad z_{ci})^T$ is the intersection between two lines perpendicular to the axes of revolution of both revolute joints $\boldsymbol{B_i}$ and $\boldsymbol{D_i}$ $\forall i = 1,..,6$ (see Fig. 9).

In Fig. 7, we can see the distance separating two points $\boldsymbol{A_i}$ and $\boldsymbol{A}_{i+2}$ evaluated by L_z $\forall i = 1,2,3,4$. Moreover, D_i, for i=1,2,5,6, are located at distance "a" from the y-axis in the platform frame.

Two mandatory assembly conditions for the mechanism to function properly are given in the following two equalities:

$$L_z \neq a, \quad \text{Here} \quad L_z \quad \text{is chosen greater than} \quad a. \tag{1}$$

$$q_i = x_i \leq x_{di} \equiv x_{di+1} \leq q_{i+1} = x_{i+1}, \forall i = 1,3,5 \tag{2}$$

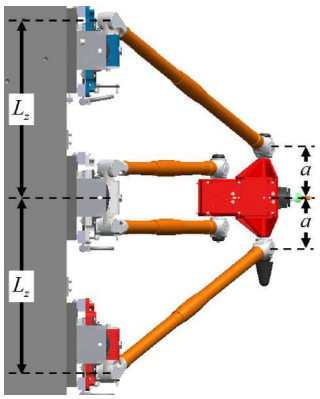


Fig. 7 Side CAD view of the parallel module of ARROW PKM. [22].

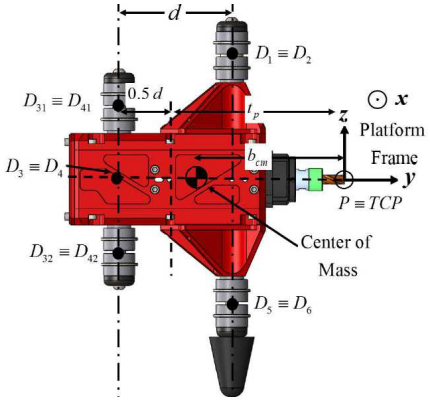


Fig. 8 Side CAD view of the traveling plate of ARROW PKM. [22].

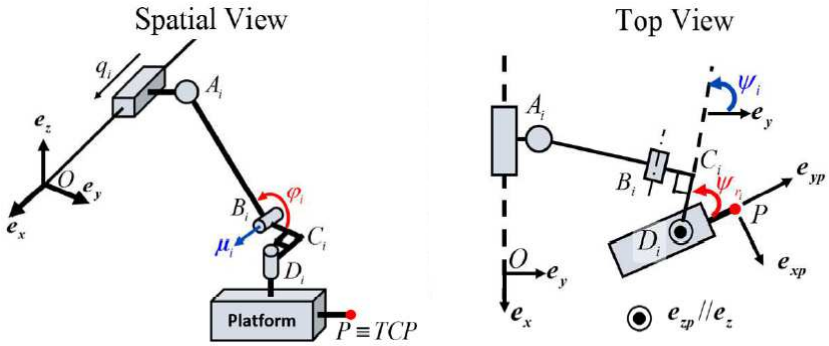


Fig. 9 ARROW PKM: schematic view of a kinematic chain with different notations.

2.2.1 Inverse Kinematic Model

Regarding the parallel module of ARROW PKM, the position and orientation of the moving platform in the 3T-1R DoFs is represented by the 4-dimensional coordinate vector $\boldsymbol{X}_{PM} = (x \quad y \quad z \quad \theta_z)^T$. The six actuators positions are parametrized by the 6-dimensional coordinate vector $\boldsymbol{q}_{PM} = (q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6)^T$. IKM represents the transformation between the cartesian space ($\boldsymbol{X}_{PM}$) and the joint space ($\boldsymbol{q}_{PM}$) for any configuration in the workspace.

Knowing the end-effector posture $\boldsymbol{X}_{PM}$ in the base frame and the coordinates of D_i in the platform frame $\boldsymbol{PD}_{\boldsymbol{i}}^{plat}$, we get $\boldsymbol{D_i}$ as follows:

$$\boldsymbol{D_i} = \boldsymbol{P} + \boldsymbol{R}(\boldsymbol{PD}_{\boldsymbol{i}}^{plat}), \quad for \quad i = 1,..,6 \tag{3}$$

where $\boldsymbol{P} = (x \quad y \quad z)^T$ is the 3T coordinates of the end-effector, and $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the basic rotation matrix of the platform frame around the z-axis. Note that $\boldsymbol{B_i}$, $\boldsymbol{C_i}$

and $\boldsymbol{D_i}$ are in the same horizontal plane (xy), whereas $\boldsymbol{A_i}$, $\boldsymbol{B_i}$ and $\boldsymbol{C_i}$ remain in the same vertical plane as illustrated in Fig. 9. Thus, $\boldsymbol{A_i^\dagger}$, the projection of $\boldsymbol{A_i}$ on the xy plane, is collinear with $\boldsymbol{B_i}$ and $\boldsymbol{C_i}$. Then, a right triangle is formed at $\boldsymbol{C_i}$ in the horizontal plane and we get the following equation:

$$||\boldsymbol{A_i^\dagger D_i}||^2 = ||\boldsymbol{A_i^\dagger C_i}||^2 + ||\boldsymbol{C_i D_i}||^2, \quad for \quad i = 1,..,6 \tag{4}$$

$||\boldsymbol{A_i^\dagger C_i}||$ can be calculated using the collinearity property of $\boldsymbol{A_i^\dagger}$, $\boldsymbol{B_i}$ and $\boldsymbol{C_i}$ as follows:

$$\begin{cases} ||\boldsymbol{A_i^\dagger C_i}|| = ||\boldsymbol{A_i^\dagger B_i}|| + ||\boldsymbol{B_i C_i}||, \quad \forall i = 1,..,6 \\ ||\boldsymbol{B_i C_i}|| = r_i, \quad \text{constant value} \\ ||\boldsymbol{A_i^\dagger B_i}|| = \sqrt{||\boldsymbol{A_i B_i}||^2 - ||\boldsymbol{A_i A_i^\dagger}||^2}, \quad \text{right triangle} \quad \overset{\triangle}{\boldsymbol{A_i A_i^\dagger B_i}} \\ \qquad\qquad = \sqrt{L_i^2 - (z_i - z_i^\dagger)^2} \\ z_i^\dagger = z_{bi} = z_{di} \end{cases} \tag{5}$$

Knowing that $||\boldsymbol{C_i D_i}|| = r_i'$ is a constant value, one can substitute (5) in (4) and get the length $||\boldsymbol{A_i^\dagger D_i}||$. Then, replace the value of $||\boldsymbol{A_i^\dagger D_i}||$ in the following equation:

$$\begin{cases} ||\boldsymbol{A_i^\dagger D_i}|| = \sqrt{(x_i^\dagger - x_{di})^2 + (y_i^\dagger - y_{di})^2 + (z_i^\dagger - z_{di})^2}, \quad \forall i = 1,..,6 \\ \qquad\qquad = \sqrt{(q_i - x_{di})^2 + (y_i - y_{di})^2} \\ z_i^\dagger = z_{bi} = z_{di}, \quad x_i^\dagger = x_i = q_i \end{cases} \tag{6}$$

Based on the assembly mode condition mentioned in (2), the IKM of ARROW PKM parallel module is derived by solving (6) and written as follows:

$$q_{PM\,i} = x_{di} + (-1)^i \sqrt{||\boldsymbol{A_i^\dagger D_i}||^2 - (y_i - y_{di})^2}, \quad \forall i = 1,..,6 \tag{7}$$

As discussed before, the turntable adds the 5th DoF to ARROW robot allowing the rotation around x-axis. IKM of the turntable delivers the joints position, knowing the value of the rotational angle is θ_x, as follows:

$$\begin{cases} q_{T\,1} = \theta_x \\ q_{T\,2} = -\theta_x \end{cases} \tag{8}$$

2.2.2 Differential Kinematic Model

The differential Kinematic model of the parallel module can be defined as a relation between the moving platform's velocity, $\dot{\boldsymbol{X}}_{PM}$, and the actuated joints velocity, $\dot{\boldsymbol{q}}_{PM}$. The matrix links both velocities known as Jacobian is derived in this paragraph.

As $\boldsymbol{A_iB_i}$ is supposed to be a rigid body doesn't change its length whatever is the configuration of the robot, we have:

$$A_iB_i{}^T \boldsymbol{v}_{A_i} = A_iB_i{}^T \boldsymbol{v}_{B_i} \quad \forall i = 1,..,6 \tag{9}$$

with the velocities of $\boldsymbol{v}_{A_i}$ and $\boldsymbol{v}_{B_i}$ being respectively:

$$\boldsymbol{v}_{A_i} = \dot{q}_i \boldsymbol{e_x} \quad \forall i = 1,..,6 \tag{10}$$

$$\begin{cases} \boldsymbol{v}_{B_i} = \boldsymbol{v}_{D_i} + \boldsymbol{w}_{hp} \times \boldsymbol{D_iB_i}, \quad \forall i = 1,..,6 \\ \boldsymbol{v}_{D_i} = \boldsymbol{v_P} + \boldsymbol{w}_{Plat} \times \boldsymbol{PD_i^{plat}} \end{cases} \tag{11}$$

where $\boldsymbol{w}_{hp} = \dot{\psi}_i \boldsymbol{e_z}$ $\forall i = 1,..,6$ and $\boldsymbol{w}_{Plat} = \dot{\theta}_i \boldsymbol{e_z}$ $\forall i = 1,..,6$ are the angular velocities of the horizontal plane $B_iC_iD_i$ and the moving platform respectively (see Fig. 9), knowing that both axes of rotation are parallel to the z-axis in the base frame. Substituting (10) and (11) in (9) yields to a relation where $\dot{\boldsymbol{q}}_{PM}$ and $\dot{\boldsymbol{X}}_{PM}$ are coupled to $\dot{\psi}$. So, there is need to find another equation that leads to uncoupled relation between $\dot{\boldsymbol{q}}_{PM}$ and $\dot{\boldsymbol{X}}_{PM}$. This equation is expressed by writing $\boldsymbol{v}_{A_i}$ in terms of $\boldsymbol{v}_{B_i}$ and $\boldsymbol{w}_{arm}$ knowing that each arm $\boldsymbol{A_iB_i}$ is performing two rotations: one around an axis parallel to the z-axis in the base frame and the other is around the axis $\boldsymbol{\mu}_i$ defined in Fig. 9. The expression will be as follows:

$$\begin{cases} \boldsymbol{v}_{A_i} = \boldsymbol{v}_{B_i} + \boldsymbol{w}_{arm} \times (-\boldsymbol{A_iB_i}), \quad \forall i = 1,..,6 \\ \boldsymbol{w}_{arm} = \boldsymbol{w}_{A_iB_i} = \dot{\varphi}_i \boldsymbol{\mu_i} + \dot{\psi}_i e_z \\ \boldsymbol{\mu_i} = \frac{\boldsymbol{r_i} \times \boldsymbol{e_z}}{\|\boldsymbol{r_i} \times \boldsymbol{e_z}\|} \end{cases} \tag{12}$$

From the relations in (9) and (12), with some computations and simplifications, the Inverse Differential Kinematic Model (IDKM) for the parallel module is derived and the form of the inverse Jacobian matrix, $\boldsymbol{J_m}$, is obtained:

$$\begin{cases} \boldsymbol{J}_{q_{PM}}\dot{\boldsymbol{q}}_{PM} = \boldsymbol{J}_{X_{PM}}\dot{\boldsymbol{X}}_{PM} \implies \dot{\boldsymbol{q}}_{PM} = \boldsymbol{J}_{\boldsymbol{m}}\dot{\boldsymbol{X}}_{PM} \quad \text{with} \quad \boldsymbol{J}_{\boldsymbol{m}} = \boldsymbol{J}_{q_{PM}}^{-1}\boldsymbol{J}_{X_{PM}} \\ \boldsymbol{J}_{q_{PM}} = \begin{pmatrix} (m_1{}^{\boldsymbol{T}}\boldsymbol{e_x})(\boldsymbol{e_z^T}(\boldsymbol{L_1}\times\boldsymbol{\mu_1})) & & \\ & \ddots & \\ & & (m_6{}^{\boldsymbol{T}}\boldsymbol{e_x})(\boldsymbol{e_z^T}(\boldsymbol{L_6}\times\boldsymbol{\mu_6})) \end{pmatrix} \\ \boldsymbol{J}_{X_{PM}} : \begin{cases} J_X(i,1) = (\boldsymbol{e_z^T}(\boldsymbol{L_i}\times\boldsymbol{\mu_i}))(\boldsymbol{e_x^T}\boldsymbol{m_i}) \\ J_X(i,2) = (\boldsymbol{e_z^T}(\boldsymbol{L_i}\times\boldsymbol{\mu_i}))(\boldsymbol{e_y^T}\boldsymbol{m_i}) \\ J_X(i,3) = (\boldsymbol{e_z^T}(\boldsymbol{n_i}\times\boldsymbol{\mu_i}))(\boldsymbol{e_z^T}\boldsymbol{m_i}) \\ J_X(i,4) = -(\boldsymbol{e_z^T}(\boldsymbol{L_i}\times\boldsymbol{\mu_i}))(\boldsymbol{e_z^T}(\boldsymbol{m_i}\times\boldsymbol{PD_i^{plat}})) \\ \boldsymbol{n_i} = \boldsymbol{A_i}\boldsymbol{C_i}, \quad \forall i = 1,..,6 \end{cases} \\ \boldsymbol{m_i} = \boldsymbol{A_i}\boldsymbol{D_i}, \quad \forall i = 1,..,6 \end{cases} \tag{13}$$

Note that $\boldsymbol{J_m}$ is not invertible in the case of redundant actuated PKMs, so the pseudo-inverse of the inverse Jacobian is used as a solution widely adopted in robotics. Thus, the Forward Differential kinematic Model (FDKM) of the parallel module is obtained as follows:

$$\dot{\boldsymbol{X}}_{PM} = \boldsymbol{J}_{\boldsymbol{m}}^{*}\dot{\boldsymbol{q}}_{PM} \tag{14}$$

where $\boldsymbol{J}_{\boldsymbol{m}}^{*} \in \mathbb{R}^{4\times 6}$ is the pseudo-inverse of the inverse Jacobian matrix.

In regard to the turntable, the IDKM can be obtained by differentiating the equations of (8) as the following:

$$\begin{cases} \dot{q}_{T_1} = \dot{\theta}_x \\ \dot{q}_{T_2} = -\dot{\theta}_x \end{cases} \tag{15}$$

2.2.3 Dynamic Model

To establish the dynamics of the parallel module of ARROW PKM, two assumptions have been taken into consideration, similar for the hypothesis applied usually on Delta-like robots, are mentioned below:

1. The dry and viscous friction forces are neglected in all passive and active joints.
2. The rotational inertia of the arms is neglected, due to their small masses in comparison with the other components. The mass of each arm is split up into two equivalent parts, one part is added to the mass of the corresponding linear actuator while the other part is considered with the moving platform.

Applying the law of motion on the linear actuators gives the following equation:

$$\begin{cases} M_a \ddot{\boldsymbol{q}}_{PM} = \boldsymbol{\Gamma}_{PM} - \boldsymbol{J}_{q_{PM}}^{\boldsymbol{T}} \boldsymbol{f} \\ \boldsymbol{M_a} = diag(m_{as}, m_{as}, m_{ap}, m_{ap}, m_{as}, m_{as}) \end{cases} \tag{16}$$

where $m_{as} \in \mathbb{R}$ is the mass including the actuator, moving cart and the half-mass of the simple arm, while $m_{ap} \in \mathbb{R}$ is the mass including the actuator, moving cart and the half-mass of the parallelogram arm. M_a is called the actuators' inertia matrix. $\boldsymbol{\Gamma}_{PM} \in \mathbb{R}^6$ is the vector representing the six actuation forces provided by the linear motors. $\boldsymbol{J}_{q_{PM}} \in \mathbb{R}^{6\times 6}$ is the joint Jacobian matrix defined in (13). $\boldsymbol{f} \in \mathbb{R}^6$ is the vector of the applied forces on the arms, at $\boldsymbol{A_i}$, resulting from the acceleration and gravitational forces acting on the platform.

The dynamics of the moving platform is described by Newton-Euler's method as follows:

$$\begin{cases} M_P \ddot{\boldsymbol{X}}_{PM} + \Lambda_c \dot{\boldsymbol{X}}_{PM} = \boldsymbol{J}_{X_{PM}}^{\boldsymbol{T}} \boldsymbol{f} + m_P \boldsymbol{g} \\ \boldsymbol{M_P} = \begin{pmatrix} m_P & 0 & 0 & -b m_P \sin(\theta_z) \\ 0 & m_P & 0 & b m_P \cos(\theta_z) \\ 0 & 0 & m_P & 0 \\ -b m_P \sin(\theta_z) & b m_P \cos(\theta_z) & 0 & I_{pzz} \end{pmatrix} \\ \boldsymbol{\Lambda_c} = \begin{pmatrix} 0 & 0 & 0 & -b m_P \cos(\theta_z) \\ 0 & 0 & 0 & -b m_P \sin(\theta_z) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{cases} \tag{17}$$

with $\boldsymbol{M_P}$ and $\boldsymbol{\Lambda_c}$ are the mass matrix and the centrifugal and Coriolis effects of the moving platform respectively. $\boldsymbol{J}_{X_{PM}} \in \mathbb{R}^{6\times 4}$ is the Cartesian Jacobian matrix defined in (13). $m_P \in \mathbb{R}$ symbolizes the total mass of the traveling plate including the added half-masses of the arms, and $\boldsymbol{g} = (0\ 0\ -9.8 m/s^2\ 0)^T$ is the gravity acceleration acting on it. The term "b" refers to the x-coordinate belonging to the center of mass of the cluster formed by the platform and the added point masses on B_i, whereas "I_{pzz}" refers to the total moment of inertia about the z-axis through TCP of the same formed cluster.

The mass m_P and the moment of inertia I_{pzz} are given as follows:

$$m_P = (mass\ of\ platform) + \frac{1}{2}\sum_{i=1}^{6} (mass\ of\ arm)_i \tag{18}$$

$$I_{pzz} = I_{platform} + \sum_{i=1}^{6} I_i \tag{19}$$

for which $I_{platform}$ is the moment of inertia about the z-axis passing through TCP only of the platform's body, and I_i is that of the point mass formed at B_i after making assumption (2) around the same axis.

Differentiating the IDKM obtained in (13) delivers the acceleration kinematic relation as: $\ddot{\boldsymbol{q}}_{PM} = \boldsymbol{J_m}\ddot{\boldsymbol{X}}_{PM} + \dot{\boldsymbol{J}}_{\boldsymbol{m}}\dot{\boldsymbol{X}}_{PM}$. By the use of this relation and equations of (16) and (17), one can formulate the Direct Dynamic Model (DDM) of the parallel module as follows:

$$\ddot{\boldsymbol{X}}_{PM} = \boldsymbol{H}\boldsymbol{\Gamma}_{PM} - \boldsymbol{\Lambda}\dot{\boldsymbol{X}}_{PM} + \boldsymbol{A_G} \tag{20}$$

where $\boldsymbol{H} \in \mathbb{R}^{4\times 6}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{A_G} \in \mathbb{R}^4$ are expressed by:

$$\begin{cases} \boldsymbol{H} = \left(\boldsymbol{M_P} + \boldsymbol{J_m^T M_a J_m}\right)^{-1} \boldsymbol{J_m^T}, \quad dim(\boldsymbol{H}) = 4\times 6 \\ \boldsymbol{\Lambda} = \boldsymbol{H M_a \dot{J}_m} + \left(\boldsymbol{M_P} + \boldsymbol{J_m^T M_a J_m}\right)^{-1} \boldsymbol{\Lambda_c}, \quad dim(\boldsymbol{\Lambda}) = 4\times 4 \\ \boldsymbol{A_G} = \left(\boldsymbol{M_P} + \boldsymbol{J_m^T M_a J_m}\right)^{-1} m_P \boldsymbol{g}, \quad dim(\boldsymbol{A_G}) = 4\times 1 \end{cases} \tag{21}$$

The Inverse Dynamic Model (IDM) of the parallel module can be reformulated from (20) by getting the actuator forces vector $\boldsymbol{\Gamma}_{PM}$ as function of the moving platform's position $\boldsymbol{X}_{PM}$ and its derivatives. In case of actuation redundancy, there is no unique $\boldsymbol{\Gamma}_{PM}$ vector for a specific desired acceleration. Considering the minimum norm solution of $\boldsymbol{\Gamma}_{PM}$ and maintaining the condition of $|\boldsymbol{\Gamma}_{PM_i}| \leq \boldsymbol{\Gamma}_{PM_{max}} \quad \forall i = 1,..,6$ (all actuators are similar) [21], the IDM of the parallel module is then arranged in joint space as follows:

$$\boldsymbol{\Gamma}_{PM} = \boldsymbol{H^*}\left(\ddot{X}_{PM} + \Lambda\dot{X}_{PM} - A_G\right) \tag{22}$$

with $\boldsymbol{H^*} \in \mathbb{R}^{6\times 4}$ being the peusdo-inverse of $\boldsymbol{H}$.

Considering the dynamics of the turntable part, the varying inertia of the object to be machined is neglected due to the low acceleration needed for the tuntable rotation. The total moment of inertia of turntable with respect to the axis of rotation is given as follows:

$$I_T = 2I_{act} + I_{txx} \tag{23}$$

where I_{act} is the rotative motor's inertia and I_{txx} is the inertia of the table rotating around its proper axis which is x-axis. Assuming that the gravitational effect is compensated for by the springs (see Fig. 5), meaning that the effect of gravity and spring cancel each other, the IDM of the turntable is given by the following equations:

$$\begin{cases} \Gamma_{T_1} = (I_T\ddot{\theta}_x)/2 \\ \Gamma_{T_2} = -\Gamma_{T_1} \end{cases} \tag{24}$$

with Γ_{T_1} and Γ_{T_2} being the torques provided by the rotative actuators driving the turntable into its rotational operation.

2.3 Singularity Analysis

In this branch, a summarized singularity analysis is stated asserting that within the feasible workspace no existence for all types of singularities [22]. Two cases have been studied: constraint and classical singularity analysis.

At first, let the desired workspace be defined as follows:

$$\begin{cases} (x) : -1.5m \leq q_i \leq +1.5m \\ (y) : -0.15m \leq y - y_0 \leq +0.15m \\ (z) : -0.15m \leq z \leq +0.15m \\ (\theta_z) : -45^\circ \leq \theta_z \leq +45^\circ \\ (\theta_x) : -45^\circ \leq \theta_x \leq +45^\circ \end{cases} \tag{25}$$

with y_0 being the y offset of the center of the desired workspace knowing that the axis of rotation of the turntable passes through $\boldsymbol{P_0} = (0 \quad y_0 \quad 0)^T$ and parallel to the x direction.

2.3.1 Constraint Singularities

Constraint singularities occur when the pair of complex chains (III) and (IV) perform undesired rotation for the platform. Normally, the angular velocity of the platform is represented by the vector $\boldsymbol{w_P} = (w_x = 0, w_y = 0, w_z)^T$. Only the two complex arms are considered in this study with modifying the position of P $\equiv$ TCP to be confounded with $\boldsymbol{D_3} \equiv \boldsymbol{D_4}$ for more simplicity. Fig. 10 shows the virtual equivalent chains with the modification, and defines new notations and subscripts ij with $i = 3,4$ and $j = 1,2$.

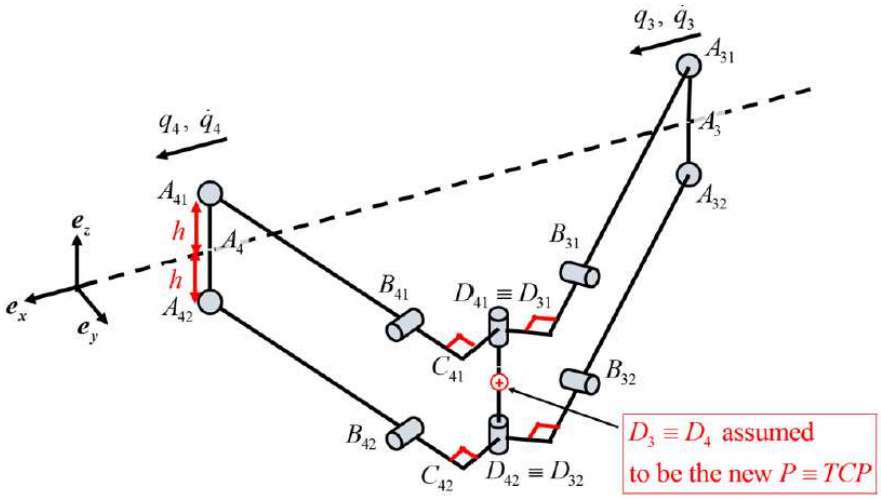
Investigating the constraint singularities starts from analysing the following equation:

$$\begin{cases} \boldsymbol{v}_{A_{ij}} = \boldsymbol{v}_{A_i} = \boldsymbol{v}_{B_{ij}} + \boldsymbol{w}_{a_{ij}} \times (-\boldsymbol{L}_{ij}), \quad \forall i = 3,4; \forall j = 1,2 \\ \boldsymbol{w}_{a_{ij}} = \boldsymbol{w_P} + \dot{\psi}_{ij}\boldsymbol{e_z} + \dot{\varphi}_{ij}\boldsymbol{\mu}_{ij} \end{cases} \tag{26}$$

Recalling that in the absence of any constraint singularity we should have $w_x = w_y = 0$ and normal generation for all necessary velocities $(v_x, v_y, w_x, w_y, \dot{\psi}_{ij}, \dot{\varphi}_{ij}, \forall i = 3,4; \forall j = 1,2)$, (26) yields to the condition below:

$$h\big(\boldsymbol{m}_{3x}\boldsymbol{m}_{4y} - \boldsymbol{m}_{3y}\boldsymbol{m}_{4x}\big)\big(\boldsymbol{L}_{3x}\boldsymbol{\mu}_{3y} - \boldsymbol{L}_{3y}\boldsymbol{\mu}_{3x}\big)\big(\boldsymbol{L}_{4x}\boldsymbol{\mu}_{4y} - \boldsymbol{L}_{4y}\boldsymbol{\mu}_{4x}\big) \neq 0 \tag{27}$$

From (27), one can distinguish three cases lead to constraint singularity:



$$
\begin{cases}
\boldsymbol{L}_{ij} = \boldsymbol{A}_{ij}\boldsymbol{B}_{ij},\ \boldsymbol{r}_{ij} = \boldsymbol{B}_{ij}\boldsymbol{C}_{ij},\ \boldsymbol{r}_{p_{ij}} = \boldsymbol{C}_{ij}\boldsymbol{D}_{ij} \\
\boldsymbol{n}_{ij} = \boldsymbol{A}_{ij}\boldsymbol{C}_{ij} = \boldsymbol{L}_{ij} + \boldsymbol{r}_{ij},\ \boldsymbol{m}_{ij} = \boldsymbol{A}_{ij}\boldsymbol{D}_{ij} = \boldsymbol{L}_{ij} + \boldsymbol{r}_{ij} + \boldsymbol{r}_{p_{ij}} \\
\boldsymbol{p}_{ij} = \boldsymbol{P}\boldsymbol{D}_{ij},\ \boldsymbol{\mu}_{ij} = \dfrac{\boldsymbol{r}_{ij} \times \boldsymbol{e}_z}{\| \boldsymbol{r}_{ij} \times \boldsymbol{e}_z \|},\ \varphi_{ij} = \left(\widehat{\boldsymbol{B}_{ij}\boldsymbol{C}_{ij}, \boldsymbol{B}_{ij}\boldsymbol{A}_{ij}}\right) = \left(\widehat{\boldsymbol{r}_{ij}, -\boldsymbol{L}_{ij}}\right) \\
\psi_{ij} = \left(\widehat{\boldsymbol{e}_y, \boldsymbol{D}_{ij}\boldsymbol{C}_{ij}}\right) = \left(\widehat{\boldsymbol{e}_y, -\boldsymbol{r}_{p_{ij}}}\right),\ \psi_{r_{ij}} = \left(\widehat{\boldsymbol{e}_{yp}, \boldsymbol{D}_{ij}\boldsymbol{C}_{ij}}\right) = \left(\widehat{\boldsymbol{e}_{yp}, -\boldsymbol{r}_{p_{ij}}}\right)
\end{cases}
$$

Fig. 10 The complex chains (III) and (IV) with notations [22].

1. $h = 0$. This is not possible because h is restricted to be opposite to zero by manufacturing process.
2. $\left(\boldsymbol{m}_{3x}\boldsymbol{m}_{4y} - \boldsymbol{m}_{3y}\boldsymbol{m}_{4x}\right) = 0$. This means the xy projections of $\boldsymbol{m_3}$ and $\boldsymbol{m_4}$ are collinear. This practically is preceded by collision between platform and slider's wall, or between the actuators. So it is on the borders or outside the accessible workspace.
3. $\left(\boldsymbol{L}_{3x}\boldsymbol{\mu}_{3y} - \boldsymbol{L}_{3y}\boldsymbol{\mu}_{3x}\right) = 0$ or $\left(\boldsymbol{L}_{4x}\boldsymbol{\mu}_{4y} - \boldsymbol{L}_{4y}\boldsymbol{\mu}_{4x}\right) = 0$. These two terms are simultaneously zero or non-zero due to the structural symmetry of the mechanism. If both are zero, this means that $\boldsymbol{L_3}//\boldsymbol{L_4}//\boldsymbol{e_z}$ which comes after a collision in the exterior region of the allowed workspace.

Therefore, the analysis outcomes guarantee that there are no constraint singularities within the operational workspace.

2.3.2 Classical Singularities

Two cases can be defined for the classical singularities to be existing: First case is described by knowing $\dot{\boldsymbol{q}}_{PM}$ but we cannot definitely determine $\dot{\boldsymbol{X}}_{PM}, \dot{\varphi} = (\dot{\varphi}_1, ..., \dot{\varphi}_6)^T$ or $\dot{\psi} = (\dot{\psi}_1, ..., \dot{\psi}_6)^T$, second case is when knowing knowing $\dot{\boldsymbol{X}}_{PM}$ but we cannot definitely determine $\dot{\boldsymbol{q}}_{PM}, \dot{\varphi}$ or $\dot{\psi}$. Mathematically, the matrices $\boldsymbol{J}_{q_{PM}}, \boldsymbol{J}_{X_{PM}}, \boldsymbol{J}_{\varphi}$ and $\boldsymbol{J}_{\psi}$ should not be rank deficient in order to avoid the aforementioned two cases. $\boldsymbol{J}_{q_{PM}}$, $\boldsymbol{J}_{X_{PM}}$ were given in (13), and $\boldsymbol{J}_{\varphi}$, $\boldsymbol{J}_{\psi}$ are derived also from equations (9) and (12) as follows:

$$\begin{cases} \boldsymbol{J}_{\varphi}\dot{\varphi} = \boldsymbol{J}_{X_{PM}\varphi}\dot{\boldsymbol{X}}_{PM} \\ \boldsymbol{J}_{\varphi} = \begin{pmatrix} (\boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1})) & & \\ & \ddots & \\ & & (\boldsymbol{e_z^T}(\boldsymbol{L_6} \times \boldsymbol{\mu_6})) \end{pmatrix} \\ \boldsymbol{J}_{X_{PM}\varphi} = \begin{pmatrix} -\boldsymbol{e_z}^T & 0 & 0 & 0 \\ -\boldsymbol{e_z}^T & 0 & 0 & 0 \\ -\boldsymbol{e_z}^T & 0 & 0 & 0 \\ -\boldsymbol{e_z}^T & 0 & 0 & 0 \end{pmatrix} \end{cases} \tag{28}$$

$$\begin{cases} \boldsymbol{J}_{\psi}\dot{\psi} = \boldsymbol{J}_{X\psi}\dot{\boldsymbol{X}}_{PM} \\ \boldsymbol{J}_{\psi} = \begin{pmatrix} (m_1{}^{\boldsymbol{T}}\boldsymbol{e_x})(\boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1})) & & \\ & \ddots & \\ & & (m_6{}^{\boldsymbol{T}}\boldsymbol{e_x})(\boldsymbol{e_z^T}(\boldsymbol{L_6} \times \boldsymbol{\mu_6})) \end{pmatrix} \\ \boldsymbol{J}_{X_{PM}\psi} = \begin{pmatrix} 0 & \boldsymbol{e_z}^T(\boldsymbol{L_1} \times \boldsymbol{\mu_1}) & -(\boldsymbol{e_x^T}\boldsymbol{\mu_1})(\boldsymbol{e_z}^T\boldsymbol{L_1}) & (e_x{}^{\boldsymbol{T}}\boldsymbol{P_1})(\boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1})) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \boldsymbol{e_z}^T(\boldsymbol{L_6} \times \boldsymbol{\mu_6}) & -(\boldsymbol{e_x^T}\boldsymbol{\mu_6})(\boldsymbol{e_z}^T\boldsymbol{L_6}) & (e_x{}^{\boldsymbol{T}}\boldsymbol{P_6})(\boldsymbol{e_z^T}(\boldsymbol{L_6} \times \boldsymbol{\mu_6})) \end{pmatrix} \\ \boldsymbol{P_i} = \boldsymbol{P}\boldsymbol{D_i^{plat}}, \quad \forall i = 1, .., 6 \end{cases} \tag{29}$$

The rank deficiencies of $\boldsymbol{J}_{q_{PM}} = \boldsymbol{J}_{\psi}$ and $\boldsymbol{J}_{\varphi}$ are given by the following singularities respectively:

$$det(\boldsymbol{J}_{q_{PM}}) = det(\boldsymbol{J}_{\psi}) = 0 \iff \exists i_0 \in \{1, .., 6\};\ \boldsymbol{e_x^T}\boldsymbol{m}_{i_0} = 0 \text{ or } \boldsymbol{e_z^T}(\boldsymbol{L}_{i_0} \times \boldsymbol{\mu}_{i_0}) = 0 \tag{30}$$

$$det(\boldsymbol{J}_{\varphi}) = 0 \iff \exists i_0 \in \{1, .., 6\};\ \boldsymbol{e_z^T}(\boldsymbol{L}_{i_0} \times \boldsymbol{\mu}_{i_0}) = 0 \tag{31}$$

Notice that (30) covers also the singularity of $\boldsymbol{J}_{\varphi}$. The case of $\boldsymbol{e_x^T}\boldsymbol{m}_{i_0} = 0$ means that m_{i_0} is perpendicular to the x-axis of the base frame which takes place when m_{i_0}

lays in the yz plane. Still this configuration is outside the regional workspace and it provokes prior collisions. For the other case, when $\boldsymbol{e_z^T}(\boldsymbol{L}_{i_0} \times \boldsymbol{\mu}_{i_0}) = 0$, then $\boldsymbol{L}_{i_0} // \boldsymbol{e_z}$ which is similar to the what have been discussed in the constraint singularities of collision between sliders and platform.

On the other hand, the rank deficiency of $\boldsymbol{J}_{X_{PM}}$ is studied after performing some linear operations achieving a simpler form. The changing of $\boldsymbol{P}$ to be confounded with $\boldsymbol{D_1}$ was for that purpose. Having a deficient rank of $\boldsymbol{J}_{X_{PM}}$ turns mathematically into the following equations:

$$\begin{cases} \boldsymbol{e_z^T}(\boldsymbol{L_3} \times \boldsymbol{\mu_3}) = \boldsymbol{e_z^T}(\boldsymbol{L_4} \times \boldsymbol{\mu_4}) = 0 \\ \text{or} \\ \boldsymbol{e_z^T}(\boldsymbol{m_3} \times \boldsymbol{r_c}) = \boldsymbol{e_z^T}(\boldsymbol{m_4} \times \boldsymbol{r_c}) = 0 \\ \text{with} \quad \boldsymbol{r_c} = \boldsymbol{D_1 D_3} \equiv \boldsymbol{D_1 D_4} \end{cases} \tag{32}$$

Or

$$\begin{cases} (\boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1}))(\boldsymbol{m_1^T e_x}) = 0 \\ \text{and} \\ (\boldsymbol{e_z^T}(\boldsymbol{L_5} \times \boldsymbol{\mu_5}))(\boldsymbol{m_5^T e_x}) = 0 \end{cases} \tag{33}$$

Or

$$\begin{cases} \boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1}) = 0 \ \text{ or } \ \boldsymbol{e_z^T}(\boldsymbol{n_5} \times \boldsymbol{\mu_5}) = 0 \ \text{ or } \ \boldsymbol{L_5^T e_z} = 0 \\ \text{and} \\ \boldsymbol{e_z^T}(\boldsymbol{L_5} \times \boldsymbol{\mu_5}) = 0 \ \text{ or } \ \boldsymbol{e_z^T}(\boldsymbol{n_1} \times \boldsymbol{\mu_1}) = 0 \ \text{ or } \ \boldsymbol{L_1^T e_z} = 0 \end{cases} \tag{34}$$

Regarding the conditions in (32), for $\boldsymbol{e_z^T}(\boldsymbol{L_3} \times \boldsymbol{\mu_3}) = \boldsymbol{e_z^T}(\boldsymbol{L_4} \times \boldsymbol{\mu_4}) = 0$ it was proved in the previous paragraph that it is not possible to occur without a prior collision. However, the second case of condition (32) (i.e $\boldsymbol{e_z^T}(\boldsymbol{m_3} \times \boldsymbol{r_c}) = \boldsymbol{e_z^T}(\boldsymbol{m_4} \times \boldsymbol{r_c}) = 0$) can be interpreted by having the four vectors $\boldsymbol{e_z}, \boldsymbol{m_3}, \boldsymbol{m_4}$ and $\boldsymbol{r_c}$ in the same plane which is either xy or yz plane. This can be take into consideration after a collision between the platform and the slider's plane or between the actuators numbered 3 and 4. Thus, it is not possible to have such condition within the operating workspace. Therefore, condition (32) has no concern with singularities inside ARROW's workspace.

Furthermore, both cases of condition (33) have been interpreted in the previous parts (specifically condition (30)).

Moreover, the possibility of having $\boldsymbol{e_z^T}(\boldsymbol{L_1} \times \boldsymbol{\mu_1}) = 0 \ \text{ or } \ \boldsymbol{e_z^T}(\boldsymbol{n_5} \times \boldsymbol{\mu_5}) = 0$ in condition (34) has been verified in the precedent parts that it is outside the executable range of the platform. For the case of $\boldsymbol{e_z^T}(\boldsymbol{L_5} \times \boldsymbol{\mu_5}) = 0 \ \text{ or } \ \boldsymbol{e_z^T}(\boldsymbol{n_1} \times \boldsymbol{\mu_1}) = 0$, it can be analysed as having the three corresponding vectors laying in the same vertical plane. Actually, we have $\boldsymbol{\mu_i} \perp \boldsymbol{n_i}$ and $\mu_{iz} = 0$, so the only incidents lead to the aforementioned condition is having $\boldsymbol{n_1} // \boldsymbol{e_z}$ or $\boldsymbol{n_5} // \boldsymbol{e_z}$ which occur after falling in collisions. Besides, the situations of $\boldsymbol{L_1^T e_z} = 0$ and $\boldsymbol{L_5^T e_z} = 0$ take place just on

the boundaries of the workspace, but it can't happen simultaneously because of the assembly condition (1) that states $L_z \neq a$. So, as well as for conditions (32) and (33), condition (34) has no possibility to lead ARROW robot falling in any singularity in the feasible workspace.

Lastly, it was shown that ARROW PKM can not fall into any type of singularities inside the functional workspace providing more flexibility and less danger of structural damaging.

3 Motion Control

In this section, the implemented control solution on ARROW robot is explained which is proportional-integral-derivative with computed feedforward. An improved PID controller is used in our case to deal with the produced internal forces due to redundancy. The integral term's overshoot and oscillations in case of saturated actuators have been avoided thanks to the anti-windup strategy added to the PID controller to be clarified in this part.

3.1 PID Control in Case of Non-Redundancy

The Proportional-Integral-Derivative (PID) control law proposed in [29] is the simplest and easiest algorithm that can be adopted for motion control of parallel robots. Although PID is non-robust against nonlinearity effects, changing parameters and uncertainties which are abundant in parallel robots, it is still the most implemented control strategy in the industrial robotized applications.

In a typical non-redundant parallel robot, the PID control law employed in joint space can be defined in discrete form as follows:

$$\begin{cases} \boldsymbol{\Gamma}_{PID}(k) = \boldsymbol{\Gamma}_P(k) + \boldsymbol{\Gamma}_I(k) + \boldsymbol{\Gamma}_D(k) \\ \boldsymbol{\Gamma}_P(k) = \boldsymbol{K_P}\tilde{\boldsymbol{q}}(k) \\ \boldsymbol{\Gamma}_I(k) = \boldsymbol{\Gamma}_I(k-1) + \boldsymbol{K_I} T_s \tilde{\boldsymbol{q}}(k) \\ \boldsymbol{\Gamma}_D(k) = \boldsymbol{K_D}\left(\frac{\tilde{\boldsymbol{q}}(k) - \tilde{\boldsymbol{q}}(k-1)}{T_s}\right) \end{cases} \tag{35}$$

where $\boldsymbol{\Gamma_P}(k)$, $\boldsymbol{\Gamma_I}(k)$ and $\boldsymbol{\Gamma_D}(k)$ are the corresponding output torques or forces of the proportional, the integral and the derivative controllers at time step k, while $\boldsymbol{K_P}$, $\boldsymbol{K_I}$ and $\boldsymbol{K_D}$ are their diagonal positive definite matrices gains respectively. $\tilde{\boldsymbol{q}} = \boldsymbol{q_d} - \boldsymbol{q}$ is the joint position error vector defined by the difference between the desired position trajectory and the measured one. T_s is the sampling period. The schematic diagram of the PID controller for non-redundant parallel manipulators is shown in Fig. 11.

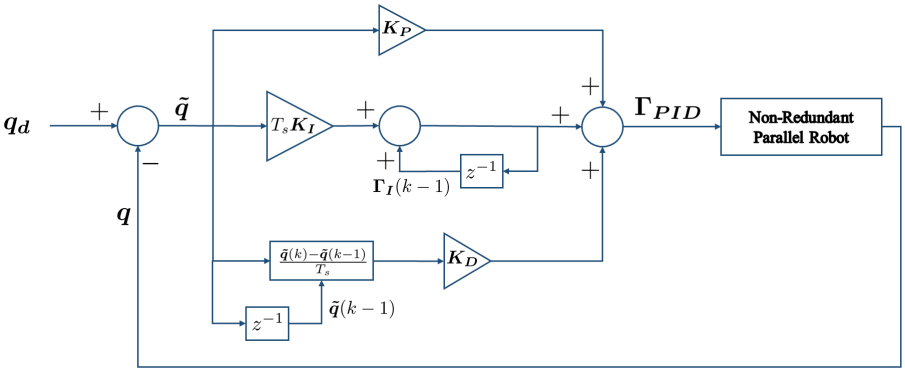


Fig. 11 Schematic view of the PID control for non-redundant parallel robot in joint space.

However, this control scheme doesn't work correctly for the redundant parallel manipulators since there are more measures than the DoFs of the moving platform, which will be discussed in the next subsection.

3.2 PID Control in Case of Redundancy

Beside the various advantageous of the actuation redundancy in parallel robots such as providing more accuracy, improved stiffness and more dexterity, it may declines the control performance of PKMs in the presence of undesired internal generated forces. These forces are mainly due to the "fight against" actuators efforts since there are more inputs than required to accomplish a specific end-effector motion.

3.2.1 Actuation Redundancy Effects

Fig. 12 shows the one DoF of the moving platform which is actuated by one actuator for the non-redundant case and by two actuators in the redundant case. In the ideal case shown in Fig. 12, the actuators and moving platform can always track the desired trajectory in both cases of redundancy and non-redundancy. Indeed, geometric errors always work out from various sources such as inaccuracies in the model geometry, measurement errors, assembly errors, non-synchronized control of the actuators, backlashes, thermal expansion, etc [15]. In the shade of those errors, applying the classical scheme of PID control shown in Fig. 11 may lead to an error with the moving platform's position tracking while still able to follow the desired actuators position signal. On the other hand, the two actuators in the redundantly actuated systems can hardly reach the desired positions at the same time conflicting to each other, which may generates antagonistic internal forces that may destroy the whole mechanism of the robot.

The position vector $\tilde{\boldsymbol{q}}$ can be considered as a vector from the actuator velocity space which always be in $C(\boldsymbol{J_m})$[3] in case of non-redundant actuated mechanisms even with the existence of geometric errors. On the contrary, in case of redundantly actuated mechanisms, $\tilde{\boldsymbol{q}}$ may has some portions from the null space and not only formed of $C(\boldsymbol{J_m})$. These errors may accumulate in the integral term of PID control generated as an input control torques or forces to the actuators. Thus the internal forces of the mechanism will be more and more significant becoming destructive to the mechanisms.

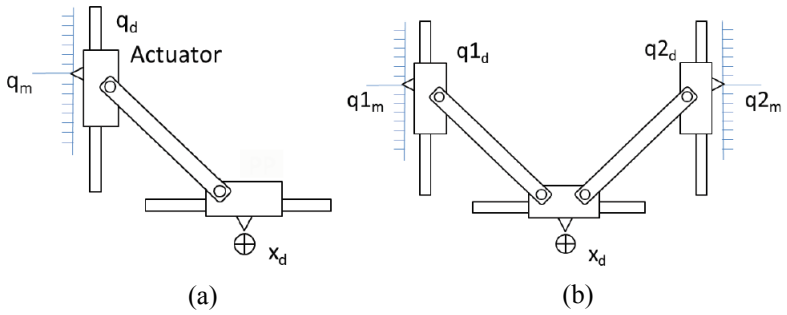
3.2.2 Adopted Solutions

Several approaches are proposed in [26] by following a general idea that is to get rid of the null space portions of $\tilde{\boldsymbol{q}}$ in the control loop. One of the propositions is to perform regularization for the input of the PID block as well as for its output.

3.2.2.1 Regularization before PID block

The regularization for the input, $\tilde{\boldsymbol{q}}$, consists of two steps described as follows:

1. Transforming the error vector from the joint space to the Cartesian space using the velocity mapping which guarantees the elimination of any residual parts out of $C(\boldsymbol{J_m})$.
$$\tilde{\boldsymbol{X}} = \boldsymbol{J_m^*}\tilde{\boldsymbol{q}} \tag{36}$$
2. Calculating again the joint space error from the Cartesian one by the inverse velocity mapping relationship, which results a regularized joint space error vector.
$$\tilde{\boldsymbol{q}}_{Reg} = \boldsymbol{J_m}\tilde{\boldsymbol{X}} \tag{37}$$



(a) (b)

Fig. 12 Actuator measurements and the moving platform's position. **(a)** Ideal non-redundant case. **(b)** Ideal redundant case [26].

[3] Columns of $\boldsymbol{J_m}$

The transformations in (36) and (37) are combined as follows:

$$\begin{cases} \tilde{\boldsymbol{q}}_{Reg} = R_V \tilde{\boldsymbol{q}} \\ \boldsymbol{R_V} = \boldsymbol{J_m} \boldsymbol{J_m^*} \end{cases} \tag{38}$$

with $\boldsymbol{R_V}$ being the regularization matrix in the velocity space.

3.2.2.2 Regularization after PID block

As it has been explained in [16], the internal forces are caused by the control inputs which are in the null space of the inverse Jacobian matrix. The regularization matrix projects the generated control inputs into the range space of $\boldsymbol{J_m}$ in two steps described as follows:

1. Using the force mapping relationship, the output forces in the actuation space are transformed to the wrench in the Cartesian space.

$$\boldsymbol{w} = \boldsymbol{J_m^T} \boldsymbol{\Gamma} \tag{39}$$

2. Calculating again the actuation space forces from the wrench by the inverse force mapping relationship, which results a regularized output forces vector.

$$\boldsymbol{\Gamma}_{Reg} = (\boldsymbol{J_m^T})^* \boldsymbol{w} \tag{40}$$

The transformations in (39) and (40) are combined as follows:

$$\begin{cases} \boldsymbol{\Gamma}_{Reg} = \boldsymbol{R_\Gamma} \boldsymbol{\Gamma} \\ \boldsymbol{R_\Gamma} = (\boldsymbol{J_m^T})^* \boldsymbol{J_m^T} \end{cases} \tag{41}$$

with $\boldsymbol{R_\Gamma}$ being the regularization matrix in the force or torque space. It was shown in [26] that the two regularization matrices in velocity and force spaces are equal, $\boldsymbol{R_\Gamma} = \boldsymbol{R_V} \equiv \boldsymbol{R_m}$.

3.2.2.3 Regularization for the integral term

The output $\boldsymbol{\Gamma}_{PID}(k)$ generated from the PID given in (42) can be shown as a linear function of the current input to PID block $\tilde{\boldsymbol{q}}(k)$, the previous input $\tilde{\boldsymbol{q}}(k-1)$, and the previous output $\boldsymbol{\Gamma}_{PID}(k-1)$. This linear relation is addressed in the following equation after several manipulations as follows:

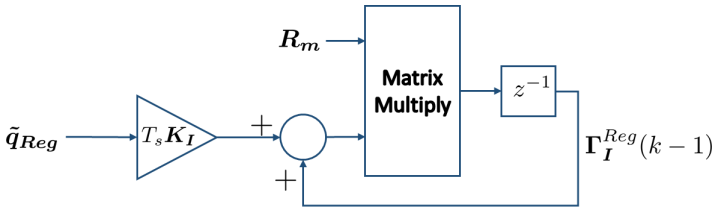


Fig. 13 Regularization algorithm for the integral term of PID controller in discrete form.

$$\begin{cases} \boldsymbol{\Gamma}_{PID}(k) = \boldsymbol{\alpha_1}\tilde{\boldsymbol{q}}(k) + \boldsymbol{\alpha_2}\tilde{\boldsymbol{q}}(k-1) + \boldsymbol{\alpha_3}\boldsymbol{\Gamma}_{PID}(k-1) \\ \boldsymbol{\alpha_1} = \boldsymbol{K_P} + T_s\boldsymbol{K_I} + \frac{1}{T_s}\boldsymbol{K_P} \\ \boldsymbol{\alpha_2} = -\frac{1}{T_s}\boldsymbol{K_D} \\ \boldsymbol{\alpha_3} = I \end{cases} \tag{42}$$

To be sure that both the input and output of PID control remain in the same subspace, we shall confirm that $\tilde{\boldsymbol{q}}(k-1)$ and $\boldsymbol{\Gamma}_{PID}(k-1)$ have the same subspace as $\tilde{\boldsymbol{q}}(k)$.

For a PID block having a regularized input position error vector, $\tilde{\boldsymbol{q}}_{Reg}$, the $C(\boldsymbol{J_m})$ may conserve its dimensions but may not stay in the same subspace during operation. This may influence the output of PID block, $\boldsymbol{\Gamma}_{PID}(k)$ may not stay always in $R(\boldsymbol{J_m})$[4]. Moreover, $\boldsymbol{\Gamma}_{PID}(k-1)$ which is calculated in a recursive way may accumulate undesired quantities in the null spaces. So there is a need to an independent regularization process on the $\boldsymbol{\Gamma}_{PID}(k-1)$ to ensure the space conservation of the input and output of PID block.

As well as the PID block with after regularization needs to a regularized $\boldsymbol{\Gamma}_{PID}(k-1)$ since the generated regularized output $\boldsymbol{\Gamma}_{Reg}$ eliminates the antagonistic forces from the mechanism but doesn't solve the problem of the increasing value of the integral term with time.

A solution for the aforementioned problems is proposed in [26] which requires a regularization for the integral term at each time step using the algorithm below (see Fig. 13):

At each step time k:

1. Regularize the generated force from the applied integration on the regularized error joint position vector $\tilde{\boldsymbol{q}}_{Reg}(k)$.

$$\boldsymbol{\Gamma}_I^{Reg}(k) = \boldsymbol{R_m}\left(\boldsymbol{\Gamma}_I^{Reg}(k-1) + \boldsymbol{K_I}T_s\tilde{\boldsymbol{q}}_{Reg}(k)\right) \tag{43}$$

2. Update the regularized value for the next time step.

$$\boldsymbol{\Gamma}_I^{Reg}(k-1) = \boldsymbol{\Gamma}_I^{Reg}(k) \tag{44}$$

[4] Rows of $\boldsymbol{J_m}$

3.2.3 Anti-Windup Strategy

To this end, it has been proposed an improved PID controller to deal with the antagonistic forces and integral term accumulation in the null space in case of redundantly actuated parallel robots. One more modification was employed to PID controller is the anti-windup strategy.

During the step change in PID controller, the integral windup occurs for saturation of actuator. Hence the system error decreases more slowly than ideal case causing the value of integral term to be increased, which may lead to significant large overshoot and settling time. This loss of performance provoked by the integral windup in systems of saturated actuators guides mostly to remove the integral term missing out its advantages in omission of the stable residual errors coming from friction or external loads.

The aim behind using the anti-windup strategy is to ensure conservation of the controller stability avoiding the oscillatory behaviour when the actuators saturated. The back-calculation approach of anti-windup strategy proposed in [7] is used here for the PID controller block. It requires feeding back the difference between saturated and unsaturated signals which may reduce the integral value. The integral output of the PID block with the three regularization treatments mentioned before and anti-windup strategy is then written as follows:

$$\begin{cases} \boldsymbol{\Gamma}_I^{Reg}(k) = \boldsymbol{R_m}\Big(\boldsymbol{\Gamma}_I^{Reg}(k-1) + \boldsymbol{K_I} T_s\big(\tilde{\boldsymbol{q}}_{Reg}(k) - \boldsymbol{K}_{AWP}\Delta\Gamma_{Reg}\big)\Big) \\ \Delta\Gamma_{Reg} = \Gamma_{Reg}^{sat} - \Gamma_{Reg} \end{cases} \tag{45}$$

where Γ_{Reg}^{sat} and Γ_{Reg} are the generated output forces or torques from the PID block after and before saturation block respectively. $\boldsymbol{K}_{AWP}$ is a diagonal positive definite matrix representing the anti-windup feedback gain. The schematic diagram illustrating the regularization technique of the integral term and the anti-windup strategy for the PID block is shown in Fig. 14.

3.3 Proposed Control Solution for ARROW PKM

PKMs are well known with their highly nonlinear dynamics which increase considerably when operating at high speed leading to instability. These nonlinearities shall be treated carefully in control algorithms in order to compensate their effects, knowing that linear controllers fail to attain this mission such as PID control.

In fact, a closed-loop control algorithm full enough with knowledge about the dynamics of the PKM will bring out good robustness against nonlinearities and disturbances. A PID control with computed FeedForward (PIDFF) is one of the most efficient non-adaptive model-based controllers that exist for parallel manipulators. PDFF control was firstly suggested in [10] as an alternative to the on-line computation requirements of others model-based motion control schemes, particularly

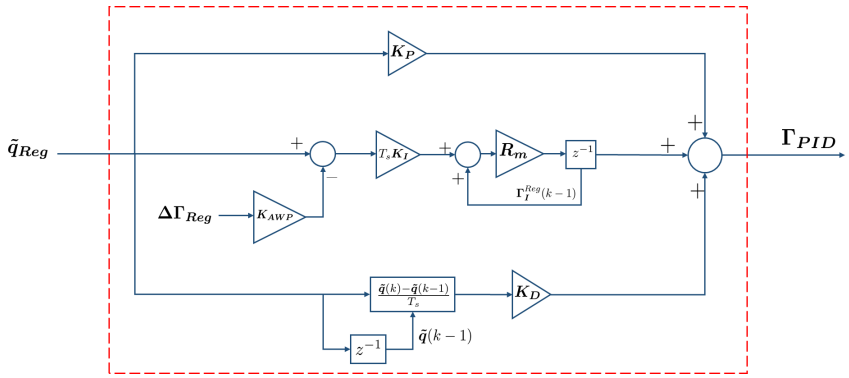


Fig. 14 PID block schematic diagram with regularized integral term and anti-windup strategy in the discrete form.

the computed torque control method. PIDFF control law consists of a linear PID feedback plus a feedforward part results from the dynamics of the robot evaluated using the desired motion trajectory. Consequently, the feedforward dynamics can be computed offline reducing the control computation efforts and making this type of model-based controllers easy and simple for implementation as well as PID controller.

For ARROW PKM, the implemented control law is the PIDFF controller using the PID block recently obtained in Fig. 14, individually for each part of ARROW: the parallel module and the turntable. The PIDFF control implemented for ARROW PKM can be written in joint space as follows:

$$\begin{cases} \boldsymbol{U}_{PM;T} = \boldsymbol{R_m}\left(\boldsymbol{K_P}\tilde{\boldsymbol{q}} + \boldsymbol{\Gamma}_I^{Reg} + \boldsymbol{K_D}\dot{\tilde{\boldsymbol{q}}}\right) + \Gamma_{PM;T\,ff} \\ \Gamma_{PM\,ff} = \boldsymbol{H_d^*}\left(\ddot{X}_d^{PM} + \Lambda_d \dot{X}_d^{PM} - A_{Gd}\right) \\ \Gamma_{T\,ff} = \begin{pmatrix} I_T/2 \\ -I_T/2 \end{pmatrix} \theta_{xd} \end{cases} \tag{46}$$

where Γ_I^{Reg} is the modified integral term including the regularization and anti-windup strategy obtained in (45). $\boldsymbol{X}_d^{PM}$, $\dot{\boldsymbol{X}}_d^{PM}$ and $\ddot{\boldsymbol{X}}_d^{PM}$ are the desired position, velocity and acceleration of the moving platform respectively. $\boldsymbol{H_d}$, $\boldsymbol{\Lambda_d}$ and $A_{G\boldsymbol{d}}$ are the matrices used in the dynamic model of the parallel module in (22) calculated based on the desired generated trajectory. $\boldsymbol{U}_{PM;T}$ is the control input forces vector applied to the linear actuators or torques vector applied to the rotative motors. θ_{xd} is the desired generated rotation in the Cartesian space around x-axis. The proposed control solution for ARROW PKM is described as a schematic view in Fig. .

It is visible in (46) that feedfoward dynamics are composed from the desired trajectories making this controller preferable for real time implementation. In other

meaning, PIDFF controller enhances the general performance and increases the accuracy with the same processing time cost of a simple PID. As well as it avoids the noise estimations of velocity and acceleration for the actuators knowing that most of the real manipulators are equipped only with position sensors (i.e encoders). Nevertheless, such kind of model-based controllers needs a precise knowledge about the dynamics of the parallel robot, which is a difficult development task, in order to achieve high level of performances.

The stability analysis of this model-based controller has been studied in the literature. In [19], it has been reported that the position error of a PD control with computed feedforward will vanish asymptotically in a local sense after selecting properly the feedback matrices. Furthermore, it has been proved in [20] that this controller is able to yield a globally asymptotically stable closed-loop system with an experimental validation.

4 Motion Generation

A strong requirement in the motion generation for a machining tool is the constant feed rate. The tool have to maintain a constant velocity along the path. To ensure this, the Gcode position data series coming from a CAD/CAM software are used as input for a Spline based definition of path.

4.1 Spline Description

We can extract, from the Gcode instructions, the tool positions along the machining path. The first idea in creating a trajectory through points is to generate straight

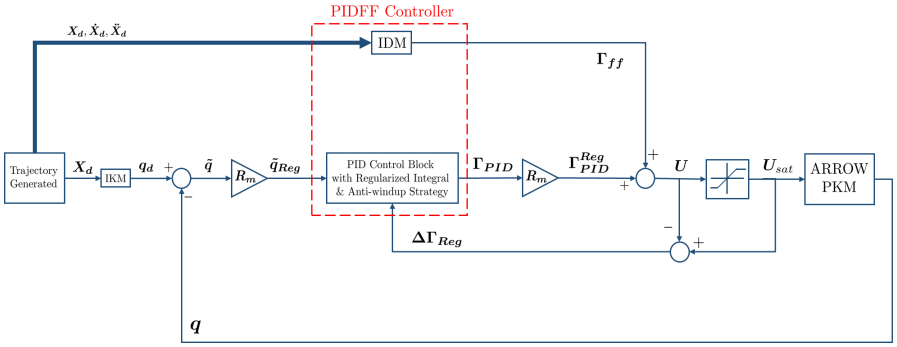


Fig. 15 PIDFF block schematic diagram with regularizations and anti-windup strategy applied on ARROW PKM.

lines between points such as in the Fig. 16. This kind of trajectory ensure a position continuity at each junction point called C_0 continuity.

Off course, this kind of trajectories will generate velocity discontinuities along the path which isn't suitable to generate a proper machining motion. A second manner to generate a trajectory is to create polynomials between points and ensure that the successive polynomials are tangent at junction points (see Fig. 17). This kind of continuity is called C_1 and is characterized by an equality of the first derivatives of successive polynomials at a junction point.

In the context of high speed machining, the continuity C1 isn't sufficient. Indeed, the path curvature from one side to the other of a point isn't continuous (ex: point 2 in Fig. 17). To ensure the continuity of curvature, called C_2 continuity, the second derivative of successive polynomials have to be equals at the junction points (ex: point 2 in the Fig. 18).

This natural cubic spline technique is used to create a path which is optimal for the machining process. A spline is a function defined piecewise by polynomials between each successive pair of points describing the path. In the case of natural cubic spline, those polynomials are of degree three with the following form:

$$P(t) = a + bt + ct^2 + dt^3 \tag{47}$$

Fig. 19 shows a spline made of four polynomials $(P_1(t), .., P_4(t))$ passing through five points. The position continuity C_0, the tangency continuity C_1 and the curvature continuity C_2 are ensured by the use of natural cubic spline curve. Warning to the parameter t which is not a time variable but a sort of spline abscissa.

In 3D space, the polynomial $P_i(t)$ is represented by a concatenated vector of three coordinate polynomials $x_i(t), y_i(t)$ and $z_i(t)$ with a similar form to (47) as follows:

$$P_i(t) = \begin{pmatrix} x_i(t) = a_{ix} + b_{ix}t + c_{ix}t^2 + d_{ix}t^3 \\ y_i(t) = a_{iy} + b_{iy}t + c_{iy}t^2 + d_{iy}t^3 \\ z_i(t) = a_{iz} + b_{iz}t + c_{iz}t^2 + d_{iz}t^3 \end{pmatrix} \tag{48}$$

The Matlab curve fitting toolbox provide the "cscvn" function allowing to get the natural cubic spline definition. This function takes as input the n Gcode path points series and give as output the spline structure. It is specified by a "break sequence"

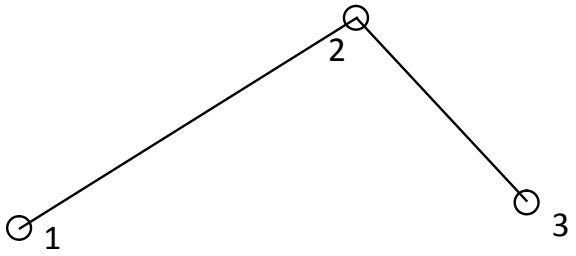


Fig. 16 Straight lines path with C_0 position continuity.

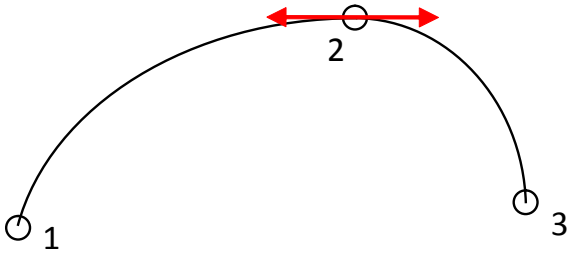


Fig. 17 Path with a tangency continuity C_1.

along the "spline abscissa" for each Gcode path point and the "coefficient array" of polynomials composing the spline shown respectively as follows:

$$breaks = [0 \quad break_1 \quad break_2 \quad ... \quad break_n] \tag{49}$$

$$coefs = \begin{pmatrix} coef\ for\ P_1 = \begin{Bmatrix} d_{1x} & c_{1x} & b_{1x} & a_{1x} \\ d_{1y} & c_{1y} & b_{1y} & a_{1y} \\ d_{1z} & c_{1z} & b_{1z} & a_{1z} \end{Bmatrix} \\ coef\ for\ P_2 = \begin{Bmatrix} d_{2x} & c_{2x} & b_{2x} & a_{2x} \\ d_{2y} & c_{2y} & b_{2y} & a_{2y} \\ d_{2z} & c_{2z} & b_{2z} & a_{2z} \end{Bmatrix} \\ \vdots \qquad\qquad\qquad \vdots \\ coef\ for\ P_n = \begin{Bmatrix} d_{nx} & c_{nx} & b_{nx} & a_{nx} \\ d_{ny} & c_{ny} & b_{ny} & a_{ny} \\ d_{nz} & c_{nz} & b_{nz} & a_{nz} \end{Bmatrix} \end{pmatrix} \tag{50}$$

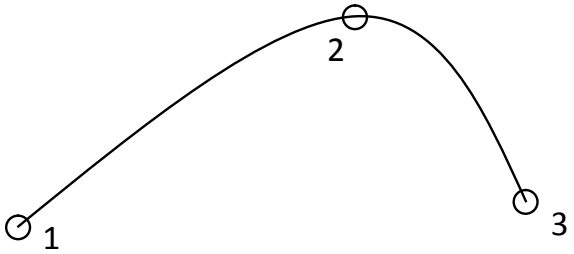


Fig. 18 Path with a curvature continuity C_2.

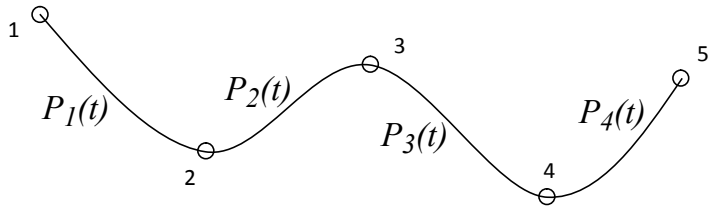


Fig. 19 Spline passing through four points (defined by four polynomials).

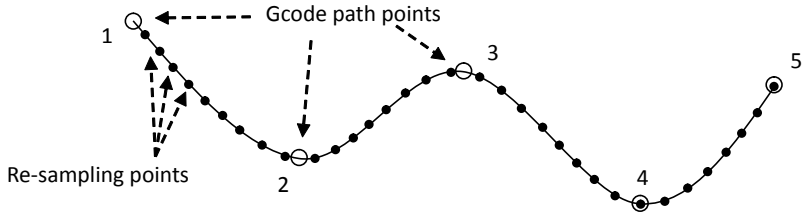


Fig. 20 Trajectory re-sampling at each sampling period.

Thus, the polynomial $P_1(t)$ give the position of Gcode point 1 when parameter $t = 0$ and gives the position of point 2 when parameter $t = break_1$. The spline $P_2(t)$ give the position of Gcode point 2 when parameter $t = 0$ and gives the position of point 3 when parameter $t = break_2$.

4.2 Spline Re-Sampling during Constant Velocity Phase

During the real time execution of the machining motion, the sampling period is fixed at $0.2\ ms$ $(5KHz)$. The Gcode path points are generated from CAD/CAM geometrical specifications and are not distributed in terms of sampling period. The spline re-sampling will consist in generating the path at each sampling period as illustrated in the Fig. 20.

As the velocity has to be constant along the path during a machining process, we have made the assumption that the distance between two successive re-sampled points is constant. Indeed, we consider that distance between two points along the path is very close to the cord distance between those two points. The determination of re-sampling points position is done by the following iterative process:

1. From Gcode point 1, small iterations δt are done on the parameter t and the corresponding position on the curve is computed from spline polynomial $P_1(t)$ as shown in Fig. 21.
2. This is repeated till the distance between the Gcode point 1 and the actual position on the curve exceed the desired distance between two successive re-sampling points.

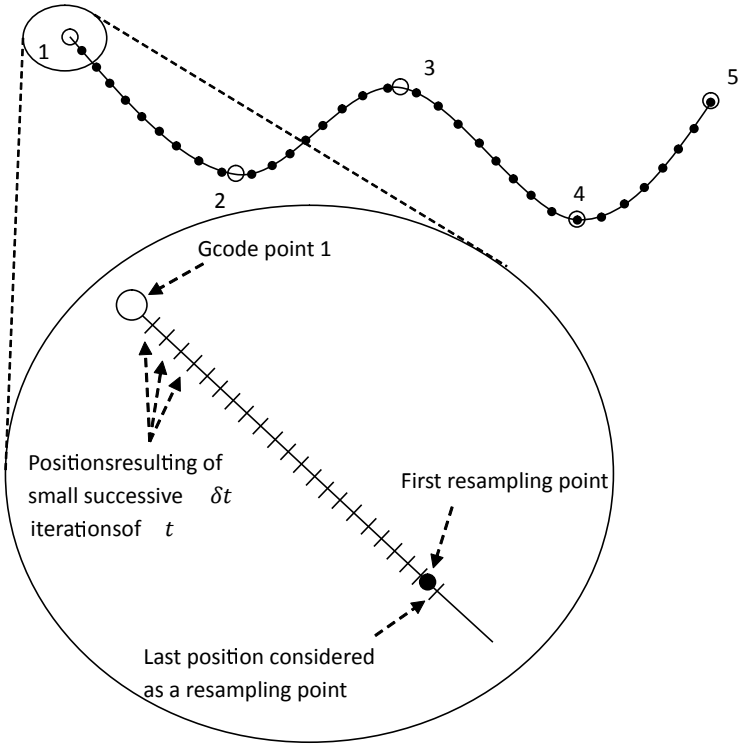


Fig. 21 Iterative process for re-sampling point determination.

3. The last position reach via this iterative process is considered as the first re-sampling point.

The process is re-iterated for the following successive re-sampling points (black points in the Fig. 21) till the reaching of Gcode point 2, 3 and so on. The more tiny δt is considered, the more precise process is obtained. During this process, the increasing of parameter t along the break sequence allow to know which polynomial has to be used to compute the positions on the curve (ex: (t) is used between Gcode point 1 and 2, $P_2(t)$ between Gcode point 2 and 3).

During the constant velocity phase, the use of spline curve guarantees an optimal trajectory in terms of velocity and acceleration discontinuities for machining motion.

4.3 Acceleration and Deceleration Phases

The previous section deals with the motion generation along the spline curve when the velocity is constant. In a fast machining process, the reaching of the constant

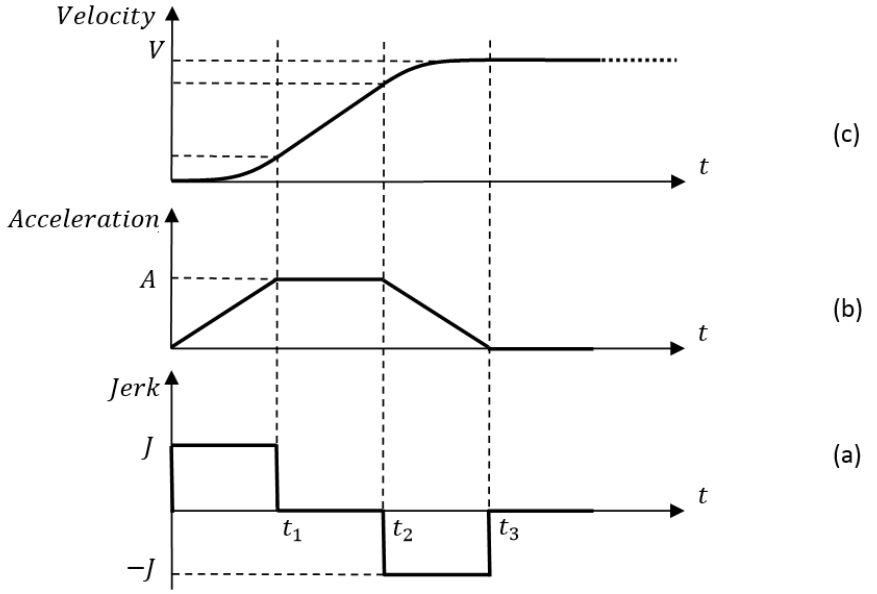


Fig. 22 Acceleration phase.

velocity phase has to be done in a smooth way aiming at increasing durability of mechanical parts. The motion has to start with an acceleration phase and to finish with a deceleration phase. A suitable acceleration procedure is used in integrating a constant jerk profile to get a proper acceleration and velocity profile. The Jerk profile (Fig. 22(a)) is introduced as follows:

$$Jerk = \begin{cases} J, & 0 < t < t_1 \\ 0, & t_1 < t < t_2 \\ -J, & t_2 < t < t_3 \end{cases} \tag{51}$$

The integration of Jerk versus time t gives the acceleration profile shown in Fig. 22(b) such as:

$$Acceleration = \begin{cases} Jt, & 0 < t < t_1 \\ A, & t_1 < t < t_2 \\ A - J(t - t_2), & t_2 < t < t_3 \end{cases} \tag{52}$$

A second integration of acceleration versus time t gives the Velocity profile shown in Fig. 22(c) such as:

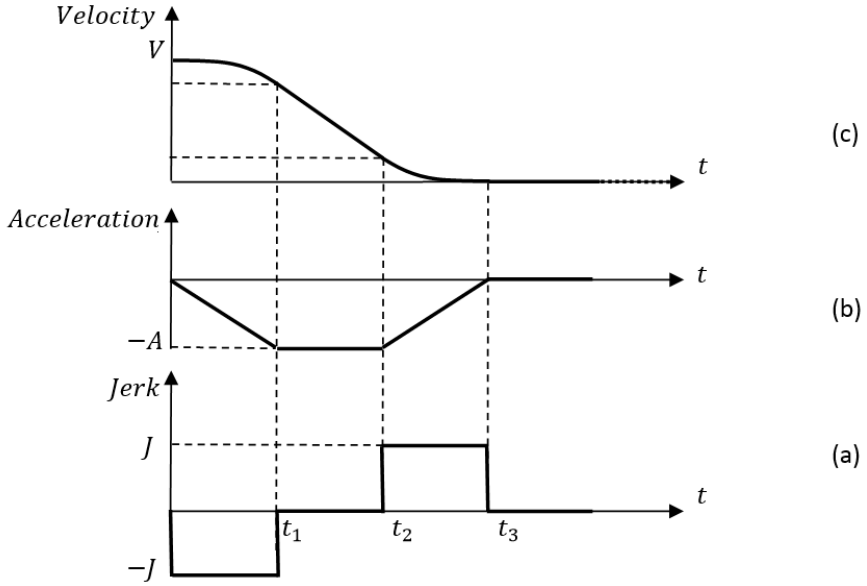


Fig. 23 Deceleration phase.

$$Velocity = \begin{cases} \frac{Jt^2}{2}, & 0 < t < t_1 \\ \frac{Jt_1^2}{2} + A(t - t_1), & t_1 < t < t_2 \\ \frac{Jt_1^2}{2} + A(t_2 - t_1) + A(t - t_2) - \frac{J(t-t_2)^2}{2}, & t_2 < t < t_3 \end{cases} \tag{53}$$

The deceleration phase is only an inversion of acceleration phase as shown in the Fig. 23. Thus, the Velocity profile can be given as follows:

$$Velocity = \begin{cases} V - \left(\frac{Jt^2}{2}\right), & 0 < t < t_1 \\ V - \left(\frac{Jt_1^2}{2} + A(t - t_1)\right), & t_1 < t < t_2 \\ V - \left(\frac{Jt_1^2}{2} + A(t_2 - t_1) + A(t - t_2) - \frac{J(t-t_2)^2}{2}\right), & t_2 < t < t_3 \end{cases} \tag{54}$$

Those velocity profiles are used in determination of re-sampling points spacing along the spline curve during acceleration and deceleration phases. Now, we have got a complete motion description at a considered sampling period.

5 Real-Time Experiments

This section exposes the real-time evaluation for the machining performance of the introduced ARROW robot. It covers the description of the experimental testbed consists of ARROW, the generated machining trajectory and the results concerning precision and performance.

5.1 Experimental Testbed of the ARROW Robot

The mechanical structure and general mechanism of ARROW PKM has been discussed previously a lot in this chapter. The manufactured ARROW robot is shown in Fig. 24, where you can see the turntable with the object to be machined, the arms and the actuators.

The parallel module actuators are all identical of type Ironless ETEL ILM12-060, providing each one a maximum force of 2500 N and reaching speed up to 15 m/s.

The actuators of the turntable are of type TMB0140-100-3RBS ETEL direct-drive motors. They can provide a maximum peak torque of 127 Nm and they are able to reach 550 rpm of maximum speed. Each actuator is equipped with a non-contact incremental optical encoder providing a total number of 5000 pulses per revolution.

The experimental setup of the ARROW robot is displayed in Fig. 25. Simulink and Real-Time Workshop from Mathworks Inc. are used to implement the control scheme and to generate the real-time execution code. The target PC is running under 5 kHz of frequency (i.e. sample time of 0.2 ms). The set of various geometric and

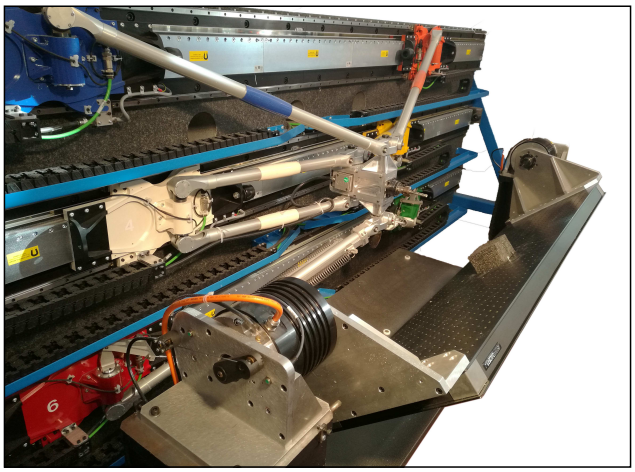


Fig. 24 The manufactured ARROW robot.

Table 1 The main dynamic parameters of ARROW robot

Parallel Module		**Turntable**	
Parameter	Value	Parameter	Value
Simple slider cart's mass	11.1 kg		
Parallelogram slider cart's mass	11.34 kg		
Simple arm linear mass	1.744 kg/m		
Parallelogram arm linear mass	3.488 kg/m	Turntable actuator's inertia	0.004 kg.m^2
Simple arms' length	0.96 m	Total inertia of the turntable	1.204 kg.m^2
Parallelograms arms' length	0.61 m		
Platform's mass	10.2 kg		
Platform's inertia	0.414 kg.m^2		

dynamic parameters of the different parts of the ARROW robot are summarized in Table 1.

The control tuning gains that are specified for both control architectures of parallel module and turntable are shown in Table 2.

The chosen machining trajectory leads to obtain the pyramid squared shape seen in Fig. 26 after the milling operation is performed on a cubed shape. A 3D plot is illustrated in Fig. from the side view of the parallel module. The chosen machined shape estimates the five DoFs of ARROW robot and allows us to evaluate the whole performance in the accessible workspace.

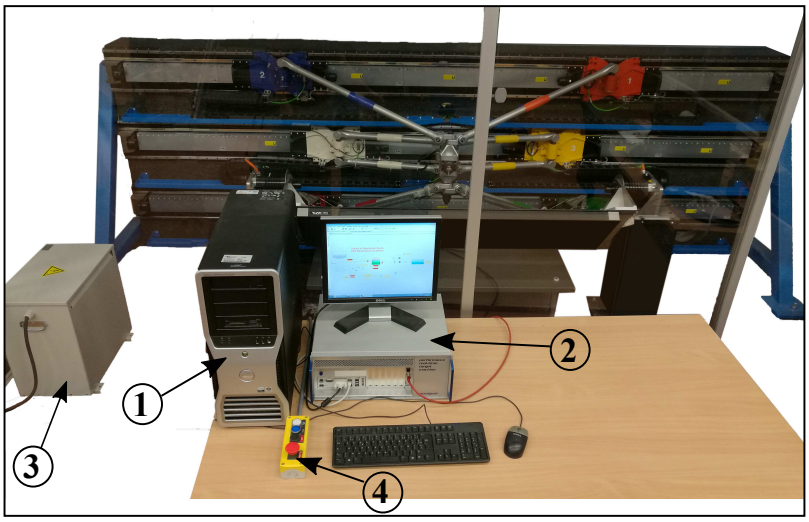


Fig. 25 The experimental setup of ARROW robot. 1: Master computer, 2: xPC Target, 3: Power supply, 4: Emergency button.

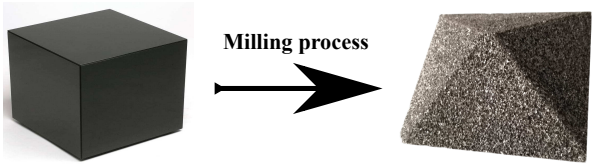


Fig. 26 The machining object before and after performing the milling process.

5.2 Experimental Results

In order to evaluate the performance of the proposed control solution, PIDFF control law, on the whole trajectory in both Joint and Cartesian spaces, the Root Mean Square of the tracking Error in Translational (RMSET) and Rotational (RMSER) motion is calculated as follows:

$$RMSET_C = \Big(\frac{1}{N}\sum_{i=1}^{N}\big(\tilde{x}^2(i)+\tilde{y}^2(i)+\tilde{z}^2(i)\big)\Big)^{1/2} \tag{55}$$

$$RMSER_C = \Big(\frac{1}{N}\sum_{i=1}^{N}\big(\tilde{\theta}_z^2(i)+\tilde{\theta}_x^2(i)\big)\Big)^{1/2} \tag{56}$$

$$RMSET_J = \Big(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{6}\tilde{q}_j^2(i)\Big)^{1/2} \tag{57}$$

$$RMSER_J = \Big(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2}\tilde{q}_j^2(i)\Big)^{1/2} \tag{58}$$

where N is the number of the time-samples.

The actual operating motion of the moving platform compared to the desired trajectory shows a high performance in terms of accuracy. Both trajectories are plotted in Fig. 28. The evolution of the tracking Cartesian error is shown in Fig. 29 zoomed to the interval [90,95] seconds for clarification purposes. The evaluation of the Cartesian tracking error is done by calculating the root mean square over the whole trajectory and is shown in Table 3. The root mean square values show a very

Table 2 Summary of the controllers' parameters

Parallel Module		**Turntable**	
Gain	Value	Gain	Value
k_P	5860796.8	k_P	5550
k_D	8070.24	k_D	25.9
k_I	20425584.8	k_I	7400
k_{AWP}	0.00166	k_{AWP}	0.000037

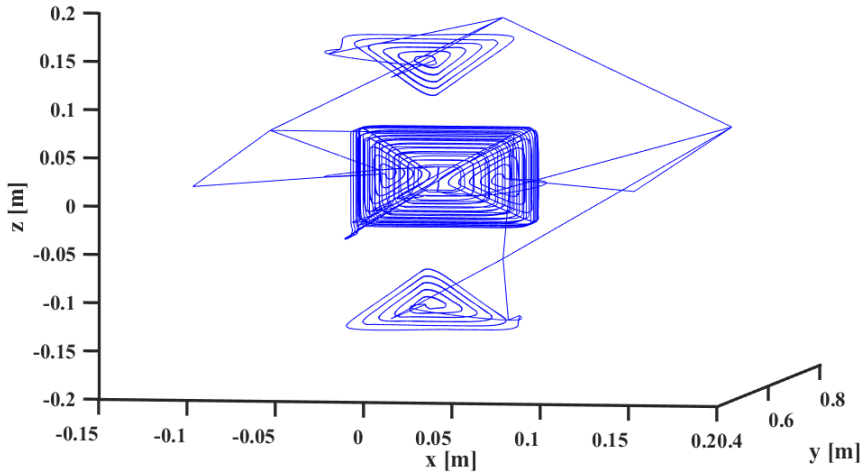


Fig. 27 3D plot for the machining trajectory as if seen from the parallel module side view.

small error over the whole trajectory which validate the high expected precision of the designed ARROW robot.

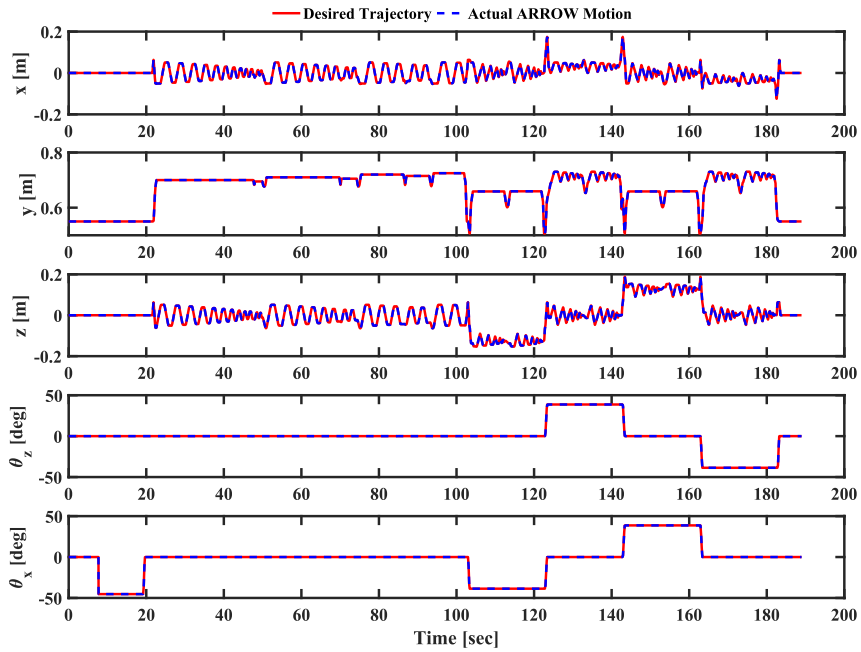


Fig. 28 Evolution of the tracking Cartesian coordinates while following the machining trajectory.

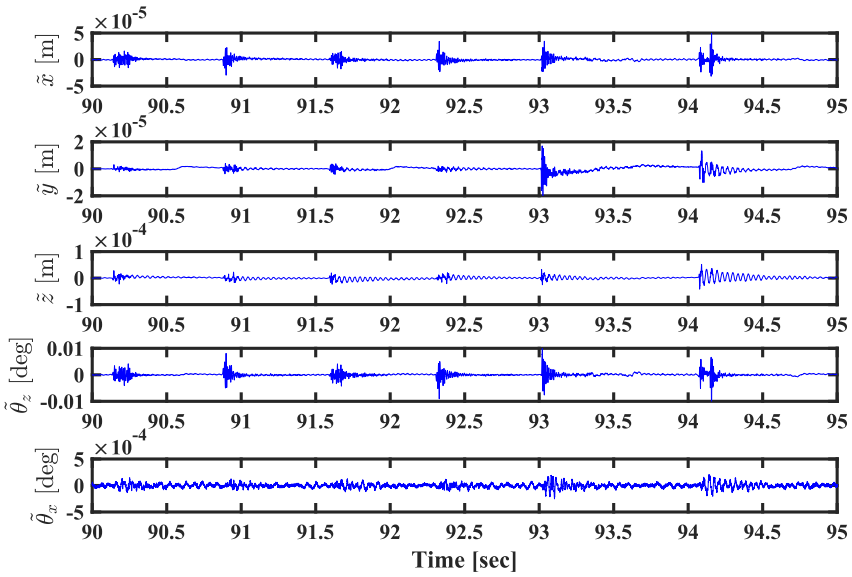


Fig. 29 Evolution of the Cartesian tracking error while following the machining trajectory.

The tracking joint errors of the parallel module and turntable are shown in Fig. 30 and Fig. 31 respectively. Also it verifies a very high performance of trajectory tracking machining task, and the evaluations of the root mean squares are mentioned in Table 3 approving the good precision. Indeed, the monitoring of Cartesian coordinates is available through measuring the joints' positions using encoders, and then transforming the obtained errors into Cartesian space by kinematic mapping (Jaccobian). This is used in the most of the parallel manipulators. As shown, both joint and Cartesian tracking errors achieved very high accuracy and that is an extra evidence to the accurate kinematic model designed for ARROW robot.

The evolution of the control input forces and torques applied to their corresponding linear motors and rotative motors are shown in Fig. 32 and Fig. 33 respectively. It is clear that all the control inputs are in the safe range of the allowed capability for the actuators. Moreover, it seems that at some instants the actuators reached the saturation zone, but the control maintained the stability and performance thanks to the anti-windup strategy that eliminates overshoots and prevent undesired oscillatory behaviour which might be dangerous for the structure.

Table 3 Root mean square of Cartesian and joint tracking errors.

Cartesian Space		**Joint Space**	
RMSET [m]	RMSER [deg]	RMSET [m]	RMSER [deg]
7.0574e-06	0.0170	6.8796e-06	0.0240

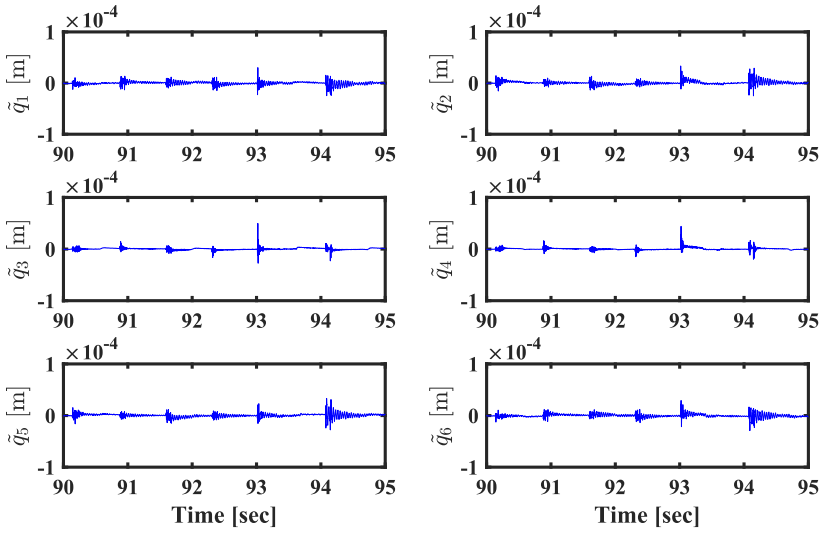


Fig. 30 Evolution of the joints tracking error of the parallel module while following the machining trajectory.

6 Conclusions and Future Work

In this work, a 5-DoFs (3T-2R) machining tool named ARROW robot based on redundantly actuated parallel structure is presented. ARROW robot is featured with a special mechanism providing a combination between rapidity and precision with large operational workspace without any type of interior singularities. A detailed explanation for the kinematics and dynamics associated to ARROW PKM were introduced in this chapter, in addition to the singularity analysis. In order to follow

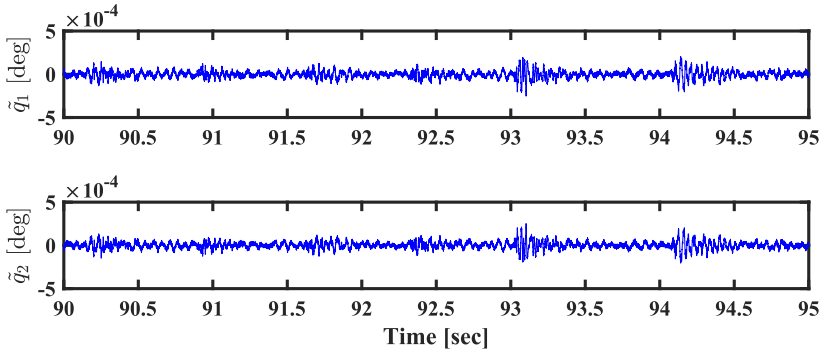


Fig. 31 Evolution of the joints tracking error of the turntable while following the machining trajectory.

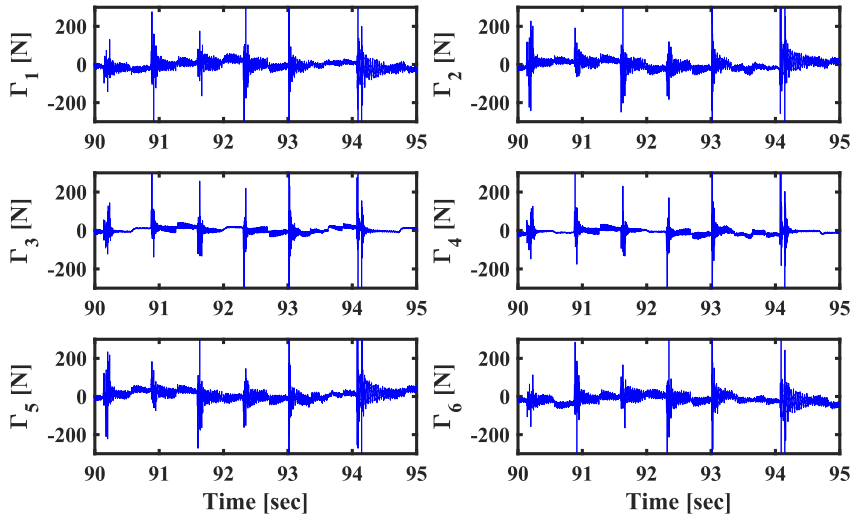


Fig. 32 Evolution of the control input forces of the parallel module linear motors.

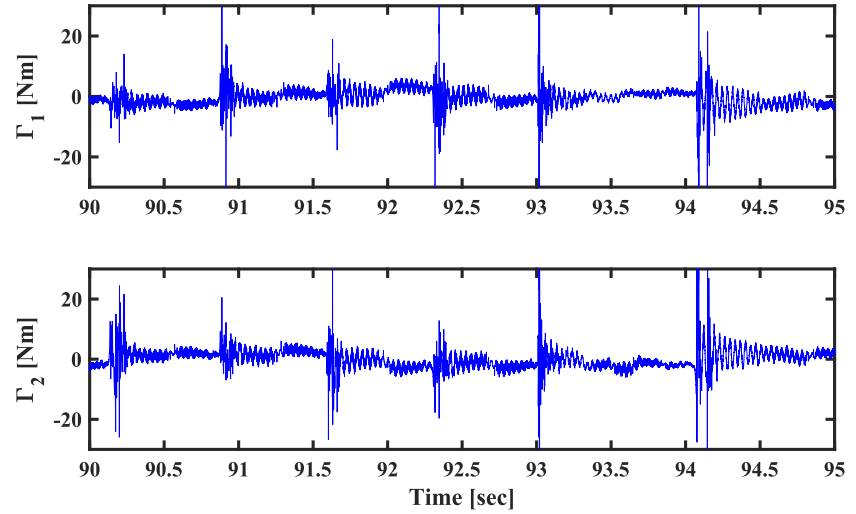


Fig. 33 Evolution of the control input torques of the turntable rotative motors.

up a machining trajectory, an improved PID with computed feedforward have been proposed as a control solution. Regularization techniques were used to eliminate the undesired antagonistic forces resulting from redundancy and to deal with the increasing value of the integral term in the PID control block. Thanks to the adopted anti-windup strategy which preserves the stability of the controller in case of saturation. Real-time experiments were conducted on ARROW PKM making a certain

machining task. The general performance was evaluated using the root mean square criteria justifying a good accuracy sufficient for a machining procedure.

For the future work, one can enhance more the control performance in terms of precision, motion speed and robustness. Implementing some adaptive model-based controllers based on the online dynamic calibration technique provides more robustness against parameters variation and disturbances. From a control point of view, considering more aspects in a PKM to be enclosed in the control-loop system can improve the general performance, such as motor drivers, actuators dynamics, transmission system and friction in the articulations.

Acknowledgements This work has been supported by the ARPE ARROW project.

References

1. Bruzzone L, Molfino R, Razzoli R (2002) Modelling and design of a parallel robot for laser-cutting applications. Innsbruck, Austria. International Conference on Modeling, Identification and Control (IASTED?02). 518–522
2. Codourey A, Honegger M, Burdet E (1997) A body-oriented method for dynamic modeling and adaptive control of fully parallel robots. Proc. 5th Symp. Robot Control, 443–450
3. Davim JP (2008) Machining: fundamentals and recent advances. Springer Science & Business Media, Verlag London, doi: 10.1007/978-1-84800-213-5
4. eFunda Machining: An introduction. In: eFunda, processes, machining. Available via DIALOG. `http://www.efunda.com/processes/machining/`
5. Grange S, Conti F, Rouiller P et al (2001) Overview of the delta haptic device. In Proc. of Eurohaptics, 5–7
6. Kerbrat O, Mognol P, Hascot JY (2011) A new DFM approach to combine machining and additive manufacturing. Computers in Industry, 62(7):684–692
7. Kumar S, Negi R (2012) A new DFM approach to combine machining and additive manufacturing. In 2nd International Conference on Power, Control and Embedded Systems.
8. Kucuk S (2012) Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization. Intech, Croatia
9. Li Y, Xu Q (2007) Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation. IEEE/ASME Transactions on Mechatronics, 12(3):265–273
10. Ligeois A, Fournier A, Aldon MJ (1980) Model reference control of high-velocity industrial robots. Proc. of the Joint Automatic Control Conference, San Francisco, CA. Session TP10 ? D no numbered pages.
11. Merlet JP (2006) Parallel Robots. 2nd ed. ser. Solid Mechanics And Its Applications, G. GLADWELL, Ed. Springer, The NetherLands
12. Bennehar M, Chemori A, Pierrot F (2014) A new extension of desired compensation adaptive control and its real-time application to redundantly actuated PKMs. Proc. of Intelligent Robots and Systems (IROS), Chicago, IL, USA, CA.
13. Bennehar M, Chemori A, Pierrot F (2014) A novel RISE-based adaptive feedforward controller for redundantly actuated parallel manipulators. Proc. of Intelligent Robots and Systems (IROS), Chicago, IL, USA, CA.
14. Bennehar M, Chemori A, Pierrot F, Creuze V (2015) Extended Model-Based Feedforward Compensation in L1 Adaptive Control for Mechanical Manipulators: Design and Experiments. Frontiers in Robotics and AI, 2:32.
15. Muller A (2009) Effects of geometric imperfections to the control of redundantly actuated parallel manipulators. In IEEE International Conference on Robotics and Automation (ICRA?09). 1782-1787

16. Muller A, Hufnagel T (2011) A projection method for the elimination of contradicting control forces in redundantly actuated pkm. In IEEE International Conference on Robotics and Automation (ICRA?11). 3218-3223
17. Natal G.S, Chemori A, Pierrot F (2012) Dual-space adaptive control of redundantly actuated parallel manipulators for extremely fast operations with load changes. IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA.
18. Natal G.S, Chemori A, Pierrot F (2015) Dual-Space Control of Extremely Fast Parallel Manipulators: Payload Changes and the 100G Experiment. IEEE Transactions on Control Systems Technology, 23(4):1520 – 1535.
19. Reyes F, Kelly R (2001) Experimental evaluation of model-based controllers on a direct-drive robot arm. Mechatronics. 11, 267-282.
20. Santibaez V, Kelly R (2000) PD control with feedforward compensation for robot manipulators: analysis and experimentation. Cambridge University Press. Robotica. 19, 11-19
21. Shayya S, Krut S, Company O, et al (2014) Dimensional synthesis of 4 dofs (3t-1r) actuatedly redundant parallel manipulator based on dual criteria: Dynamics and precision. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14). 1716-1723
22. Shayya S (2015) Towards Rapid and Precise Parallel Kinematic Machines. PhD thesis, Universit Montpellier (Ex UM2)
23. Shoham M, Burman M, Zehavi E et al (2003) Bone-Mounted Miniature Robot for Surgical Procedures: Concept and Clinical Applications. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 19(5):893–901
24. Stewart D (1965) A platform with six degrees of freedom. ARCHIVE: Proceedings of the Institution of Mechanical Engineers, 180:371–386
25. Su Y, Duan B, Zheng C (2004) Nonlinear pid control of a six-dof parallel manipulator. IEE Proceedings-Control Theory and Applications, 151:95–102
26. Yang H (2012) Agile mobile manufacturing for large workpieces. PhD thesis, Universit Montpellier (Ex UM2)
27. Youssef HA, El-Hofy H (2008) Machining technology: machine tools and operations. Taylor & Francis
28. Zhang YX, Cong S, Shang WW et al (2007) Modeling, identification and control of a redundant planar 2-dof parallel manipulator. International Journal of Control, Automation and Systems, 5:559–569
29. Ziegler J, Nichols N (1942) Optimum settings for automatic controllers. Transactions of the American Society of Mechanical Engineer, 64:759–768